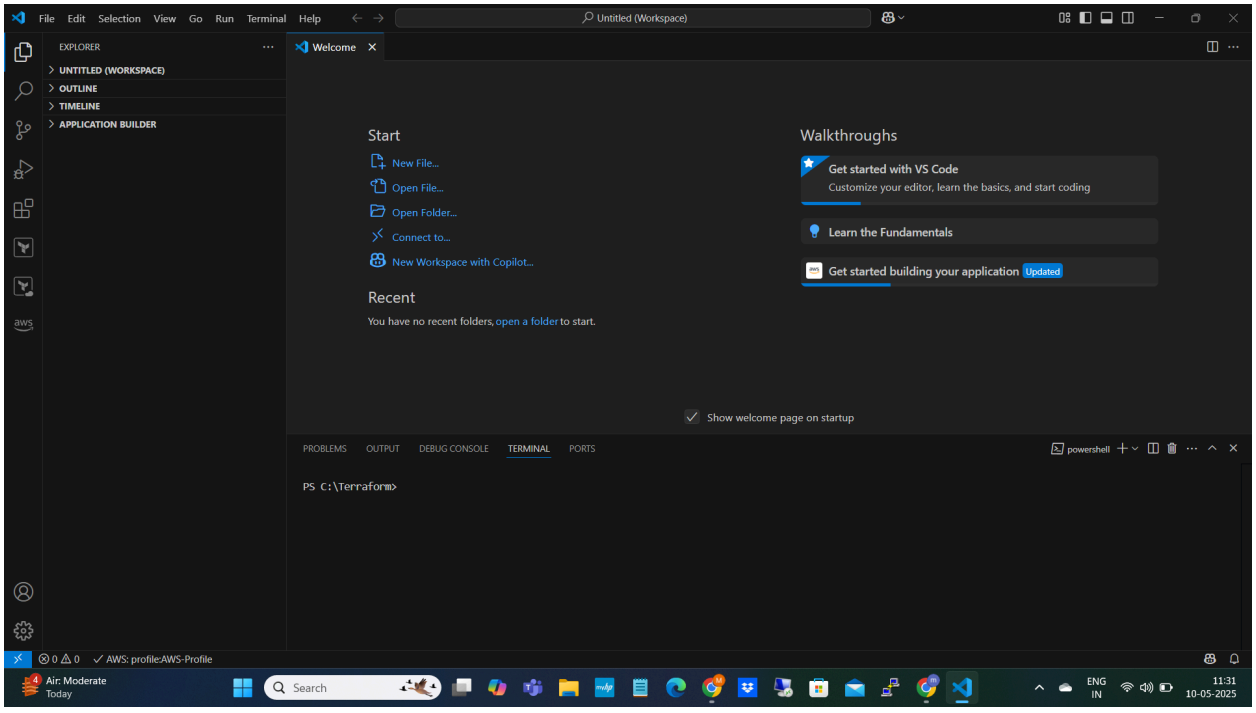
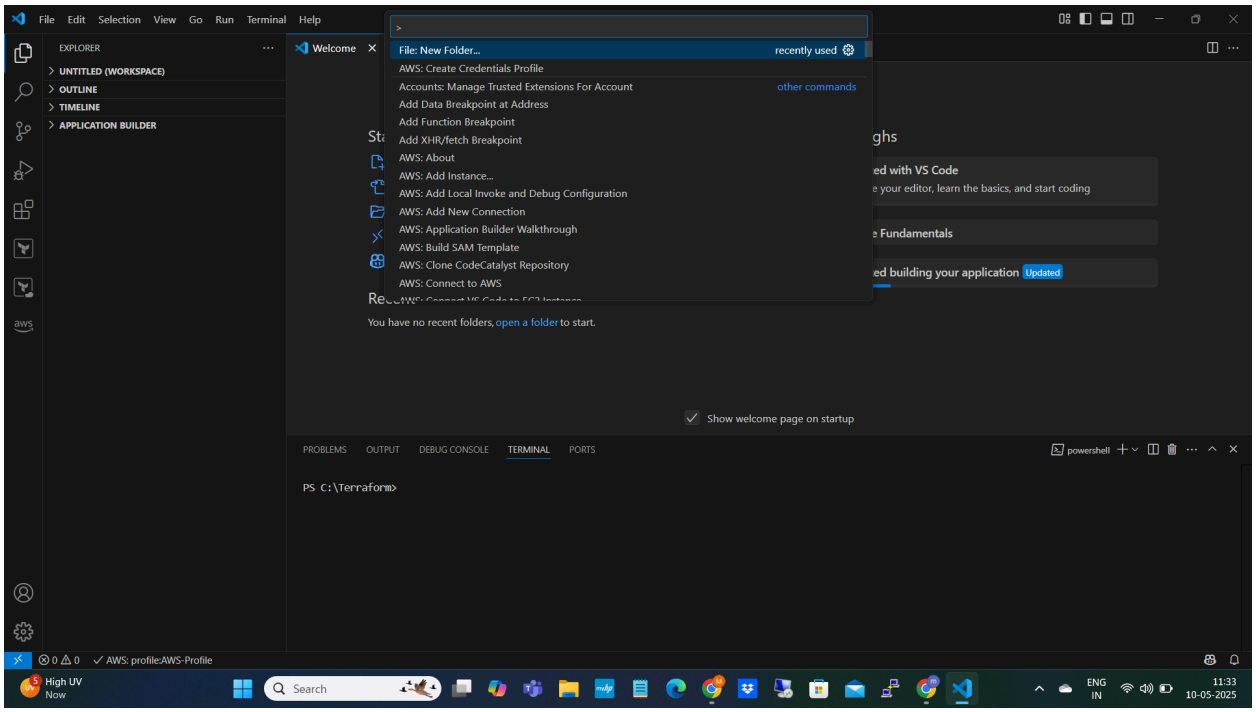
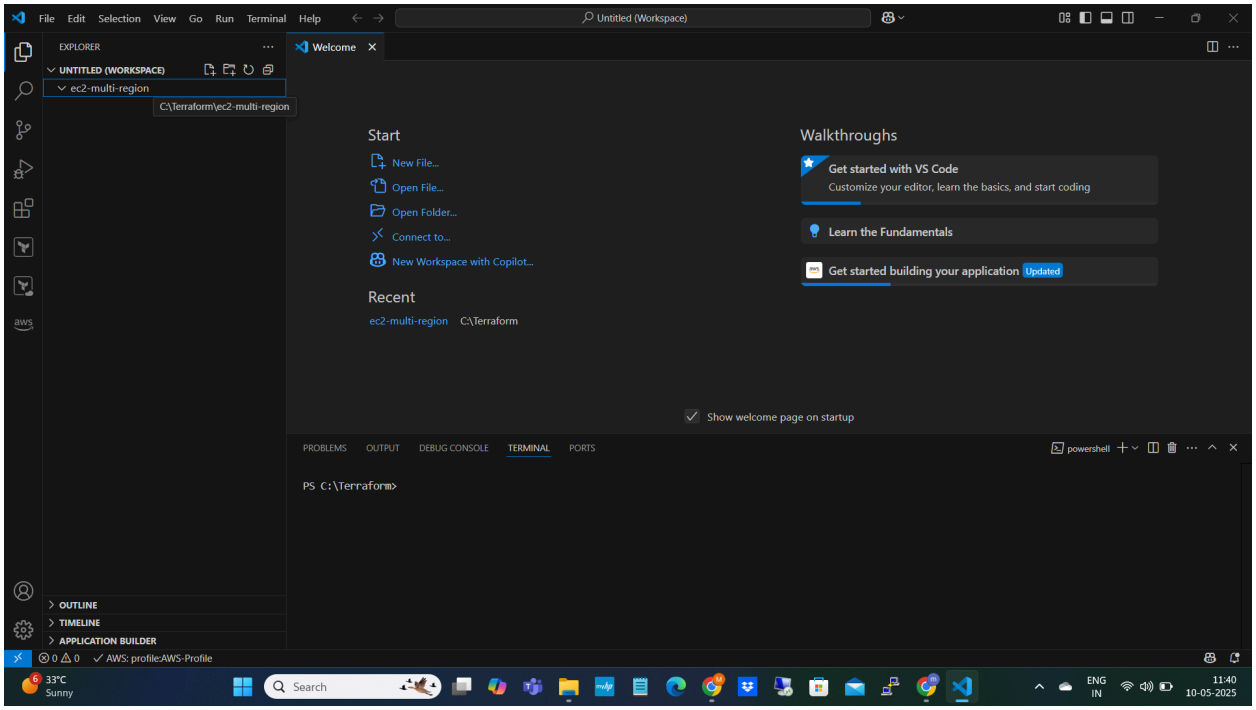
Terraform Task-2

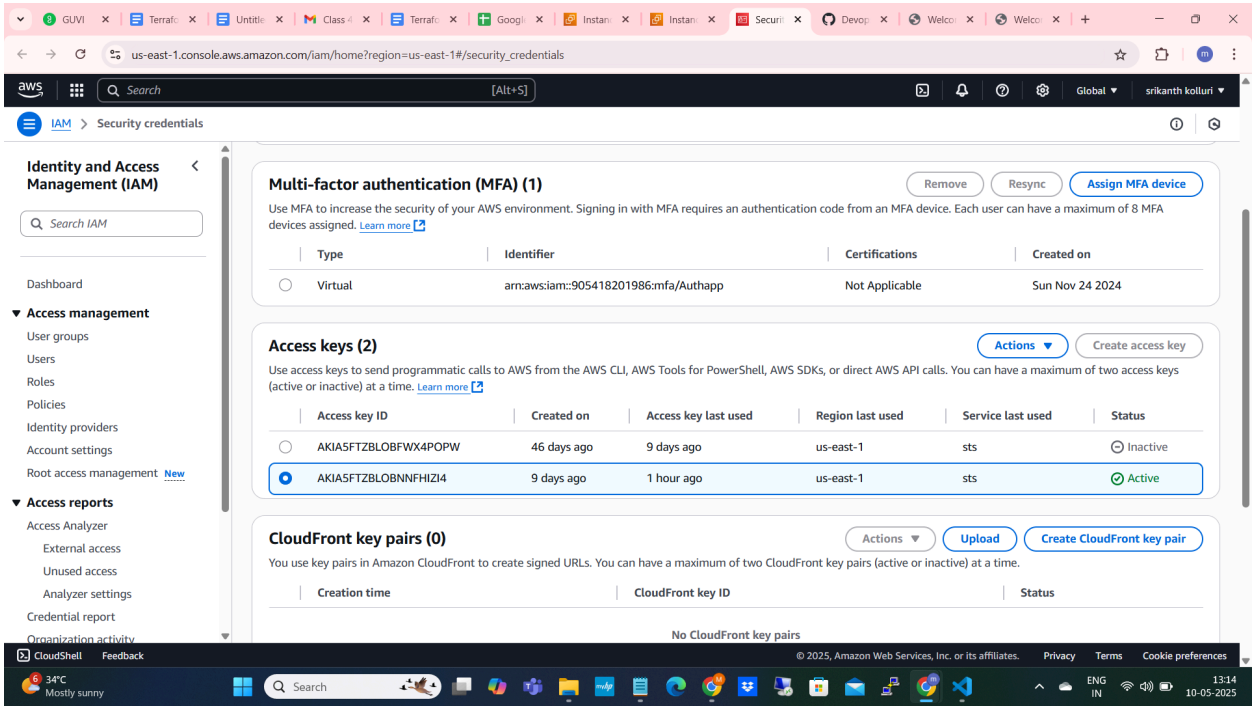
Creating Folder in Visual studio to create terraform files
Folder: **ec2-multi-region**





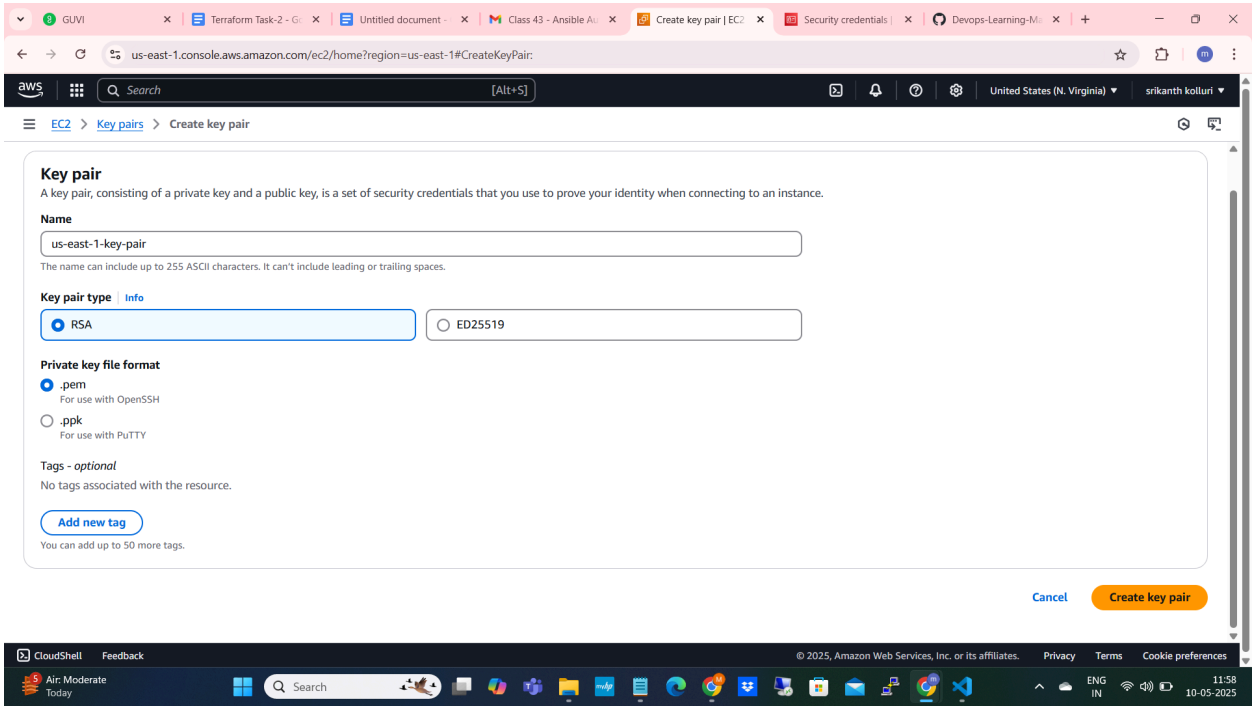


To Create 2 EC2 Instances required 2 key pairs in 2 different regions and Access Key **Accesskey:**

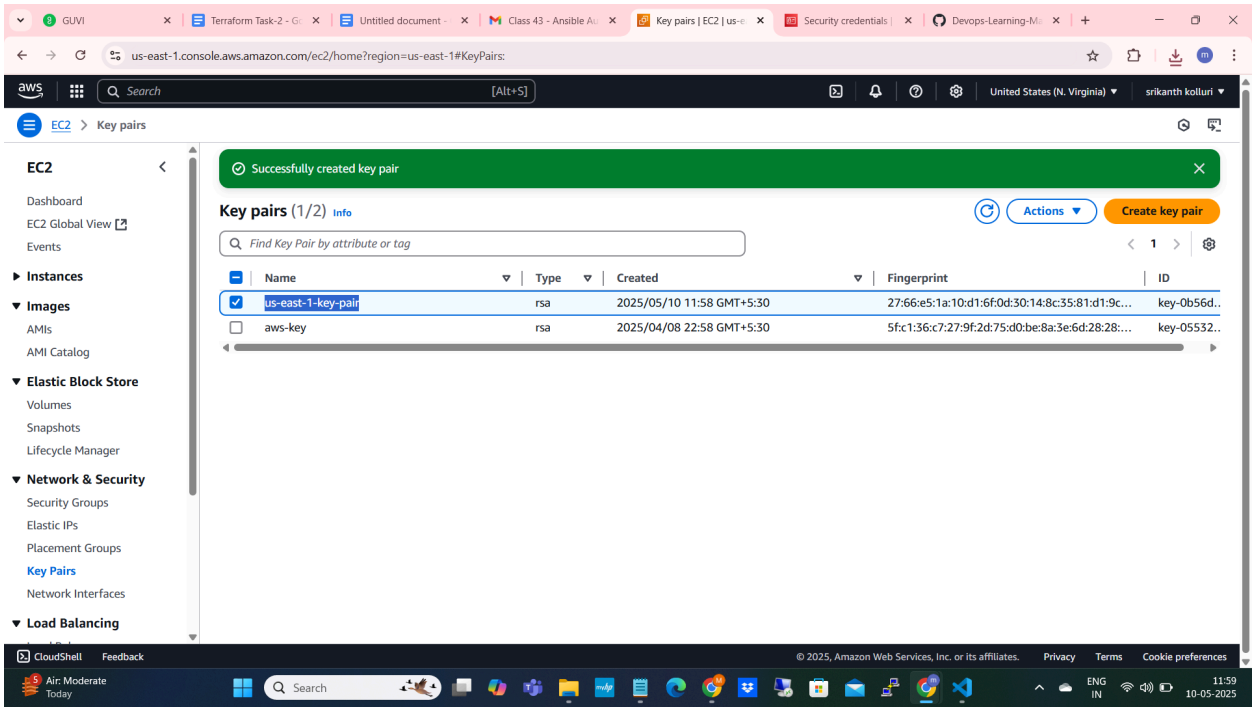


Creating 2 Key pairs for 2 regions in AWS Account:

1. Us-east-1: us-east-1-key-pair

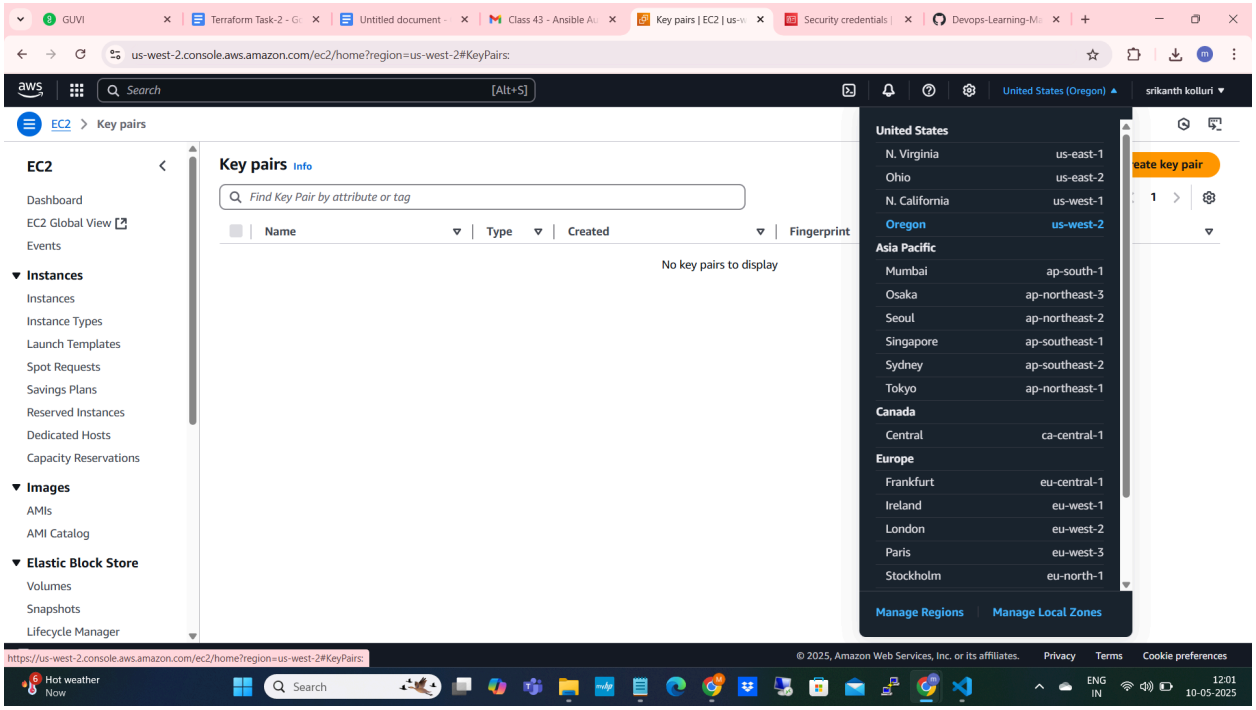


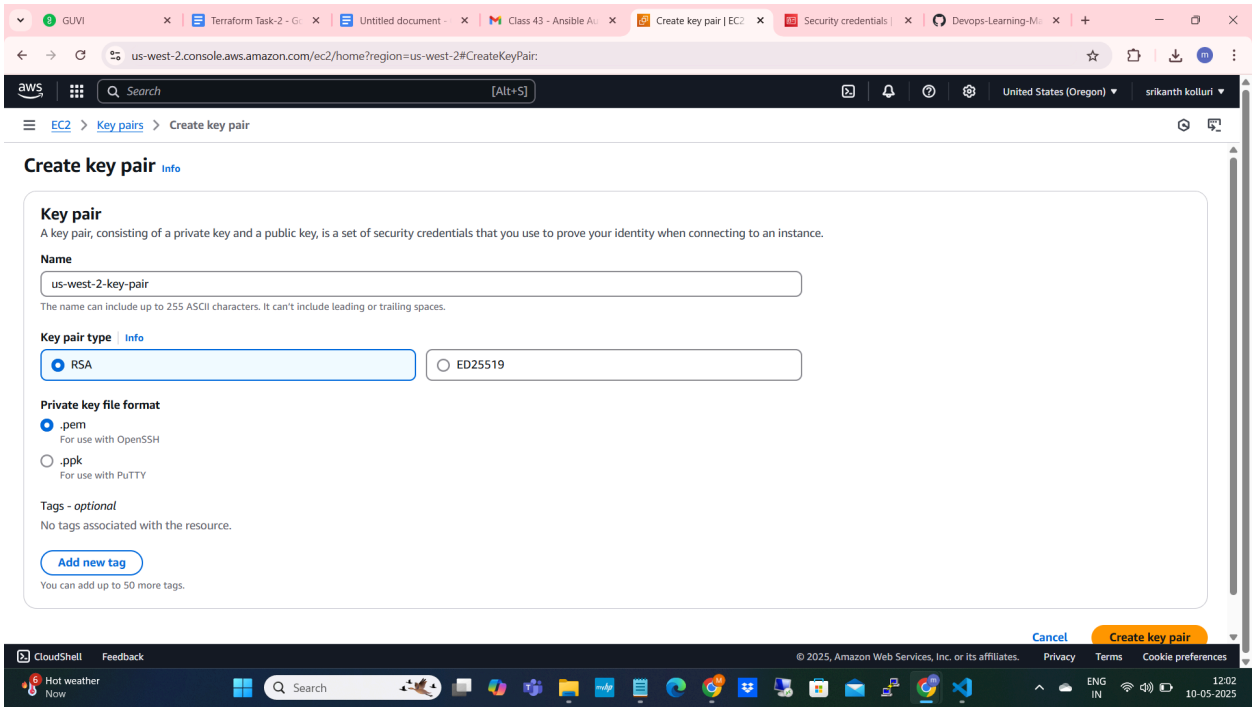
Click on Create Key pair and it is created successfully



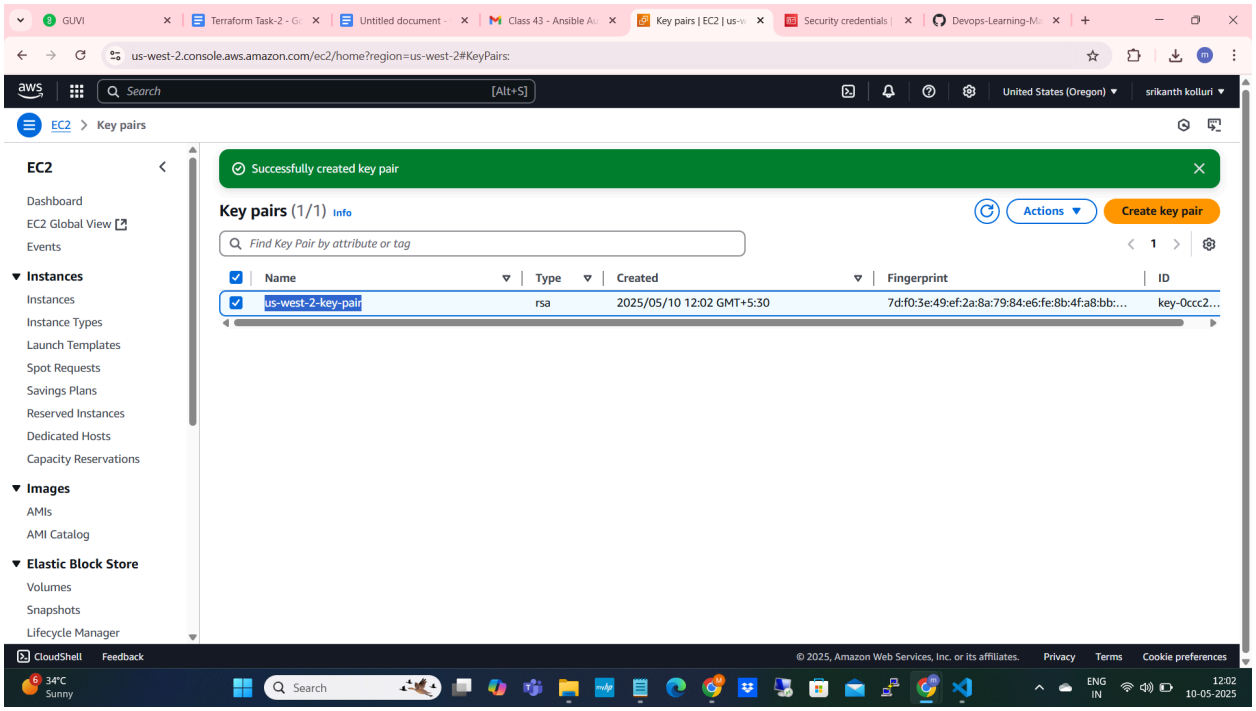
Changed the region and creating the new key pair

2. Us-west-2: us-west-2-key-pair

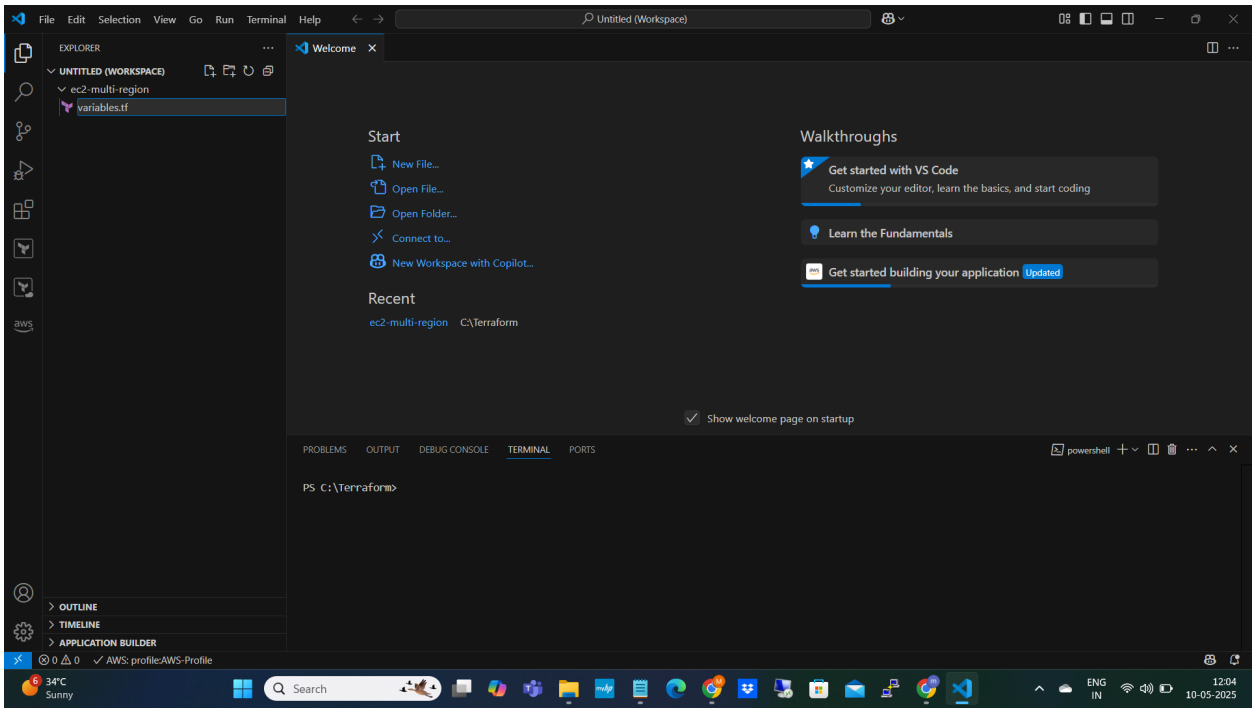




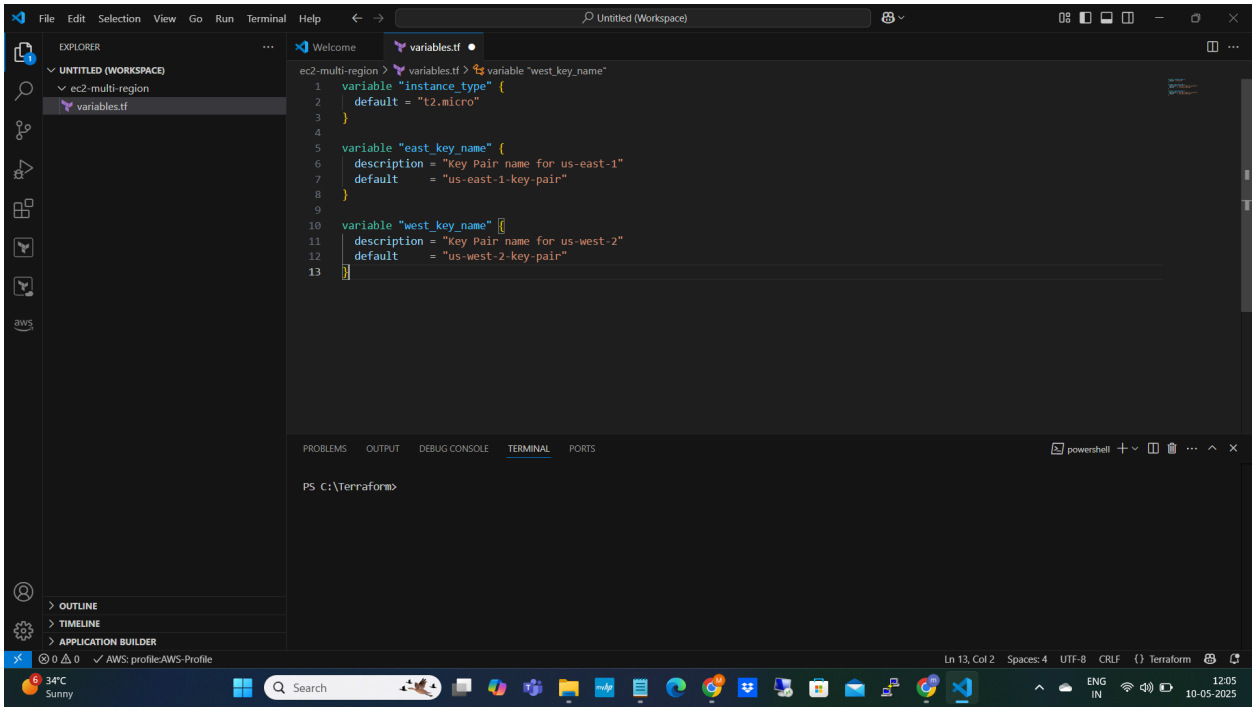
Click on Create Key pair and it is created successfully



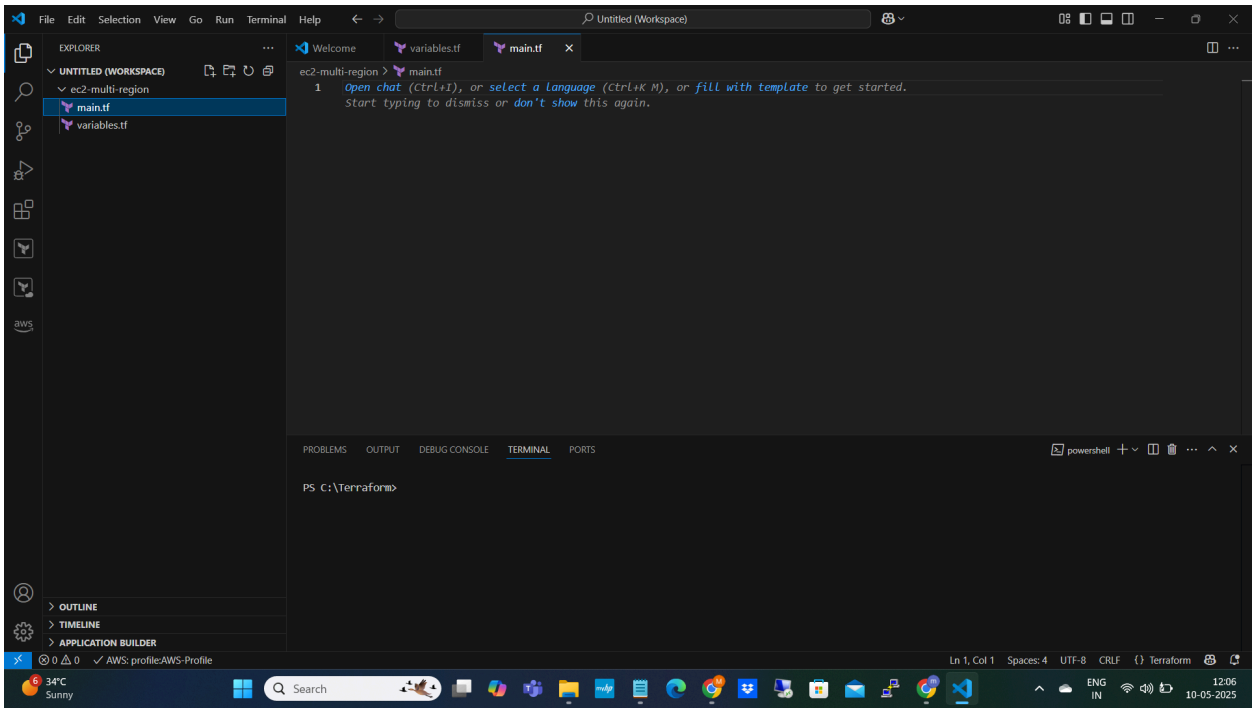
Now Create the variables.tf File inside the folder ec2-multi-region



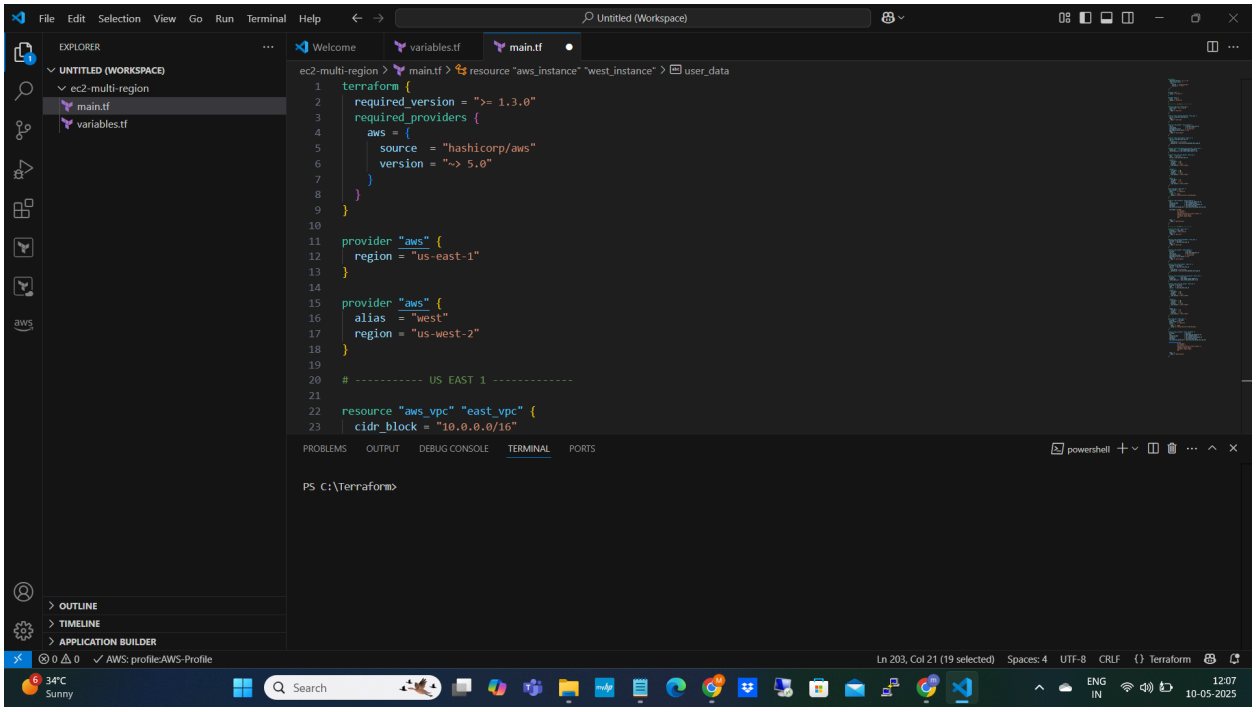
Define the Variables:



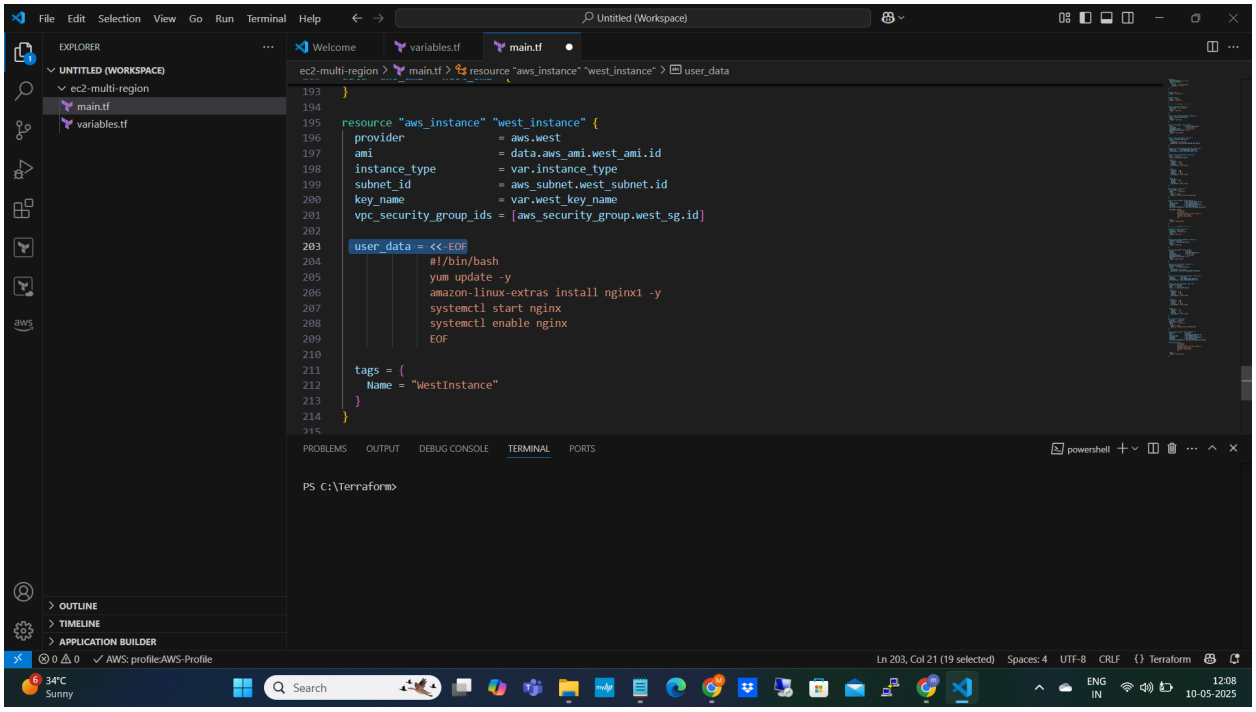
Create the main.tf file



Provide the terraform script to Create 2 EC2 instance in 2 regions and installing Nginx on them

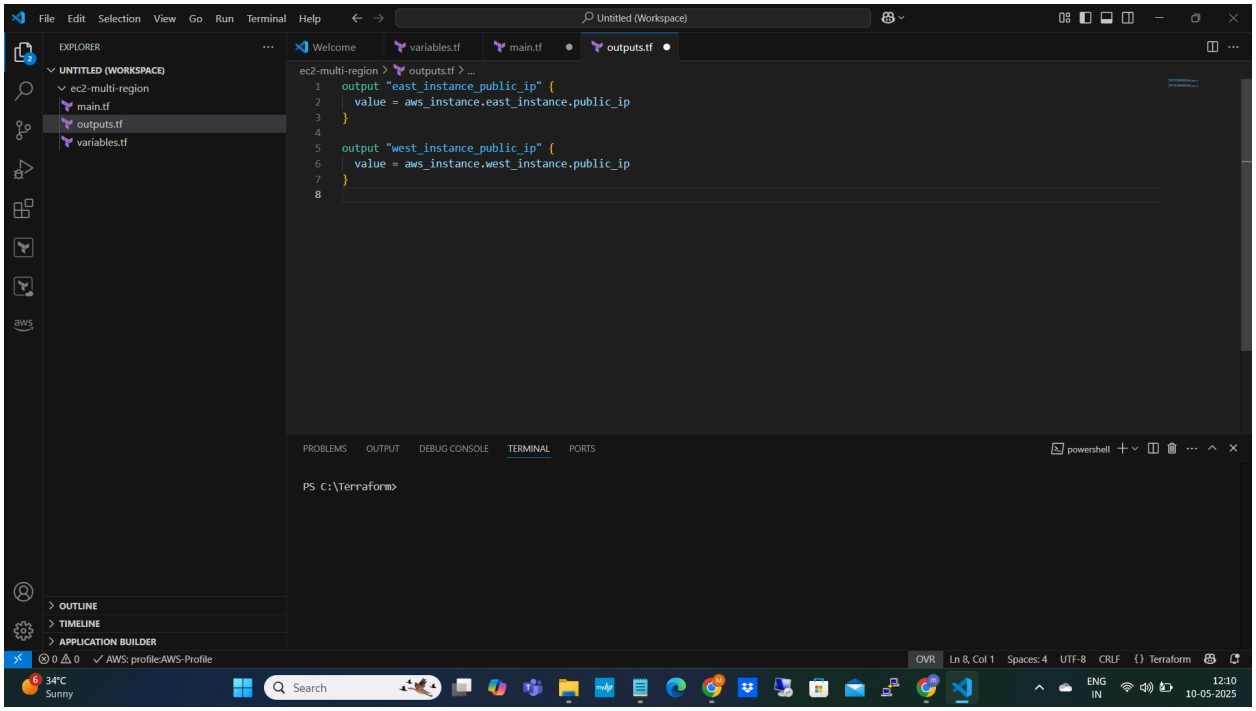


```
terraform {
  required_version = ">= 1.3.0"
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = "us-east-1"
}

provider "aws" {
  alias  = "west"
  region = "us-west-2"
}

# ---------- US EAST 1 ------------

resource "aws_vpc" "east_vpc" {
  cidr_block = "10.0.0.0/16"
```



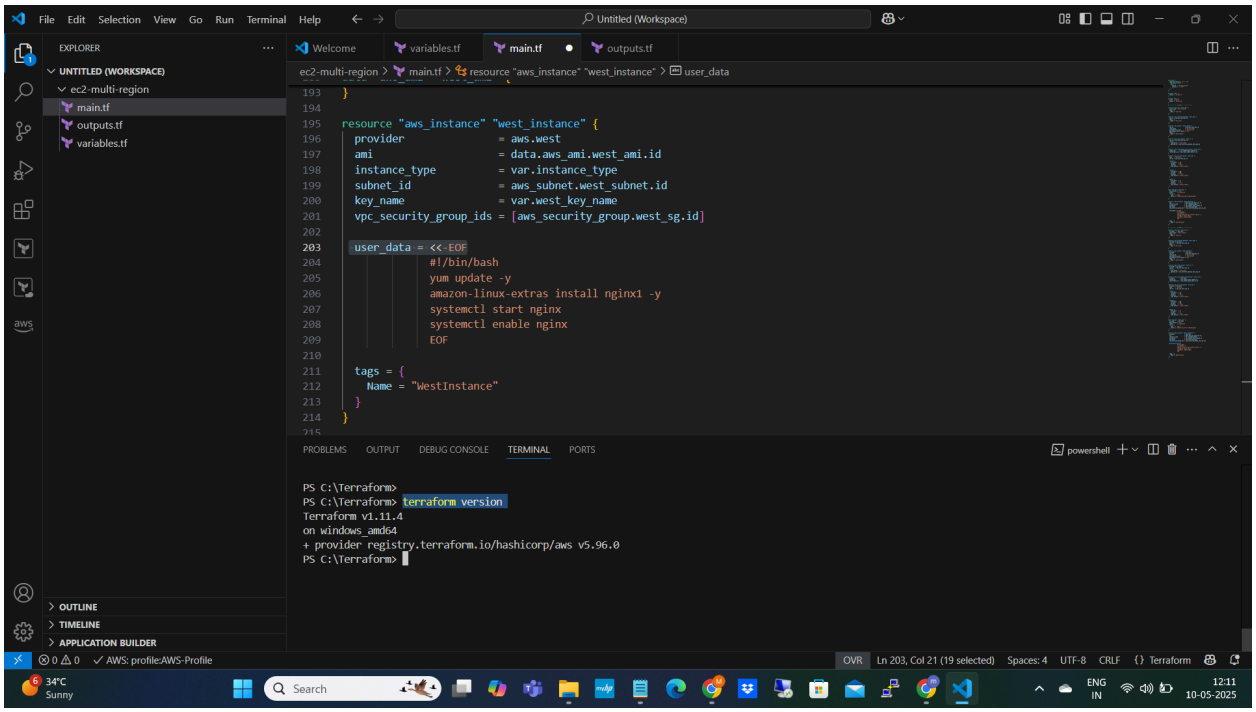
```
}

resource "aws_instance" "west_instance" {
  provider               = aws.west
  ami                    = data.aws_ami.west_ami.id
  instance_type          = var.instance_type
  subnet_id              = aws_subnet.west_subnet.id
  key_name               = var.west_key_name
  vpc_security_group_ids = [aws_security_group.west_sg.id]

  user_data = <<-EOF
              #!/bin/bash
              yum update -y
              amazon-linux-extras install nginx1 -y
              systemctl start nginx
              systemctl enable nginx
              EOF

  tags = {
    Name = "WestInstance"
  }
}
```

Create output.tf file:

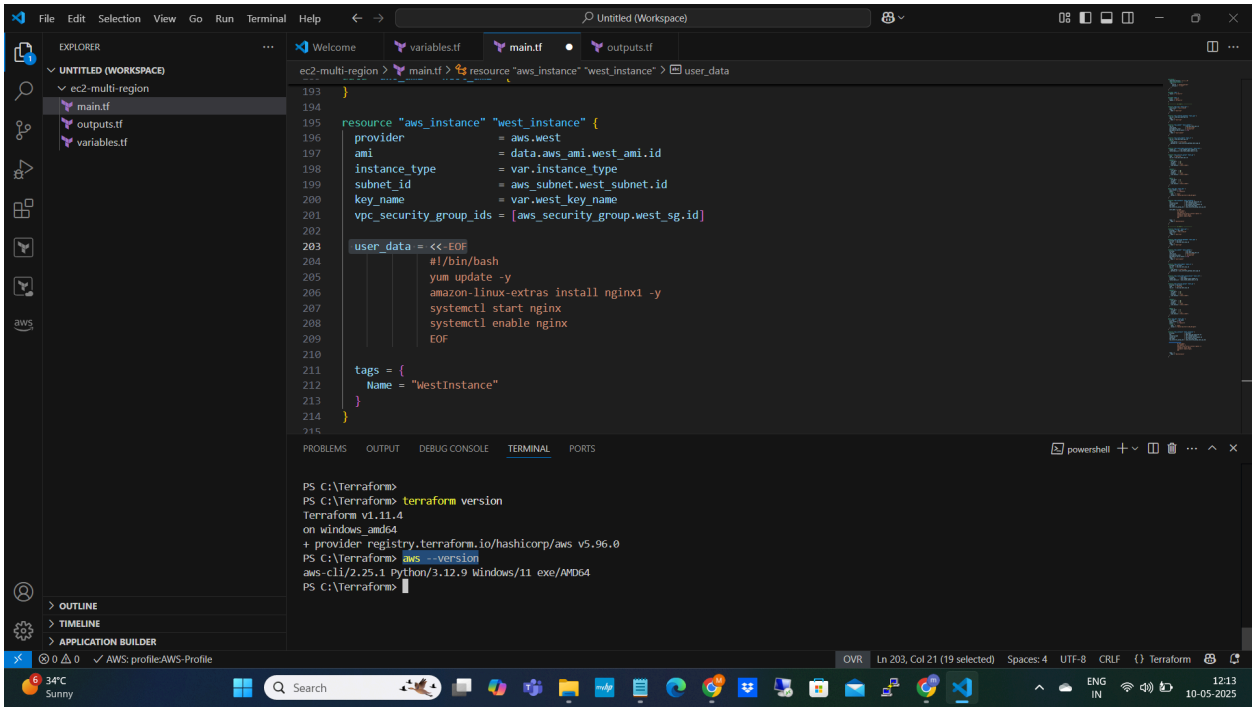


Terraform Already installed in my system

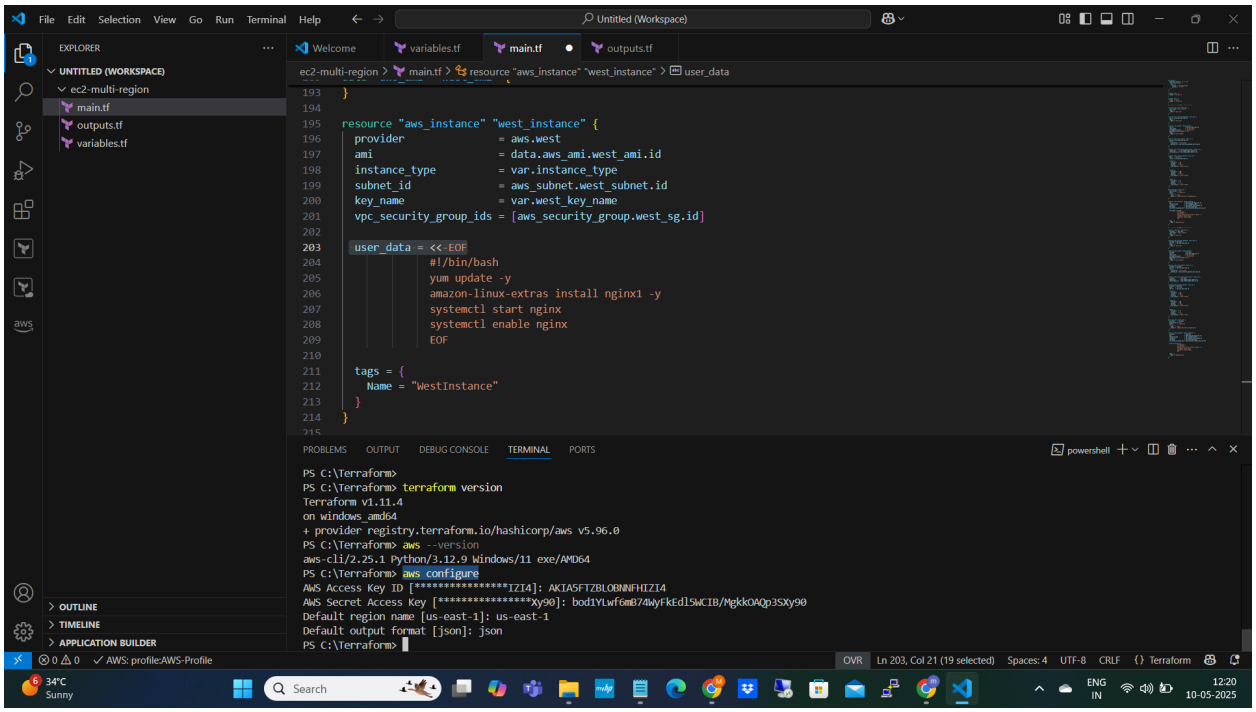
Let's check the version: **$ terraform version**



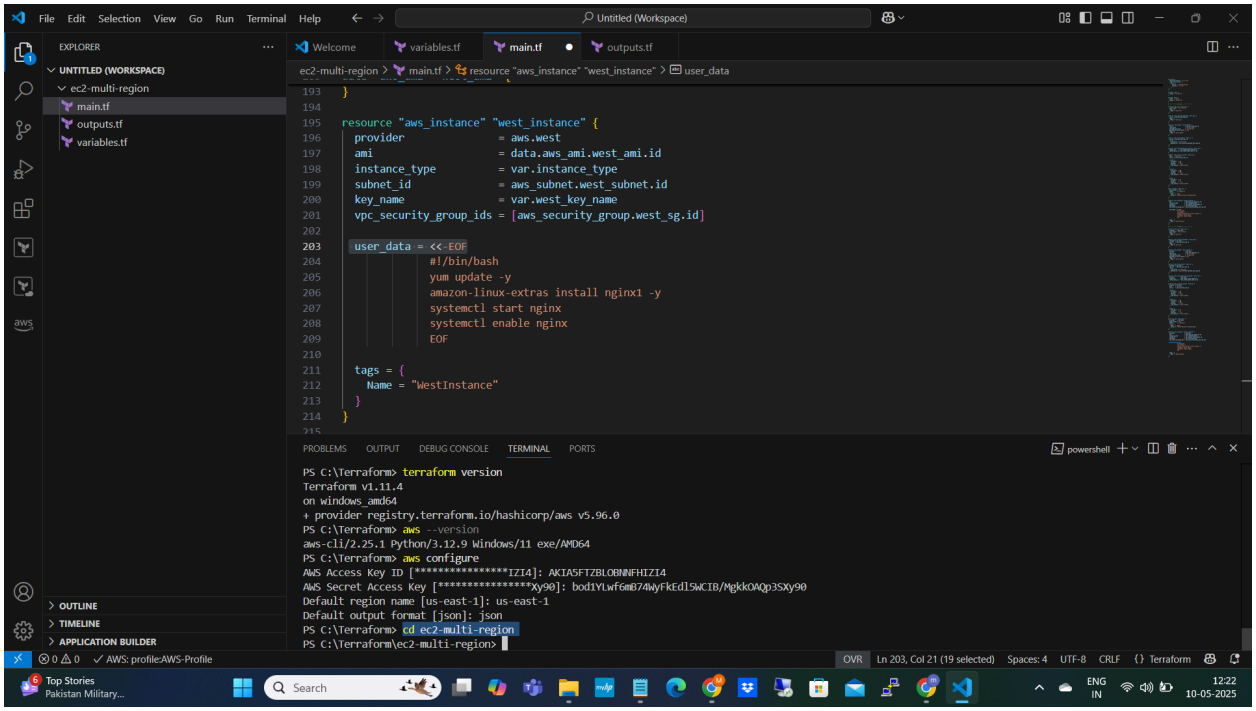
Now Configure the Aws : $ **aws --version**



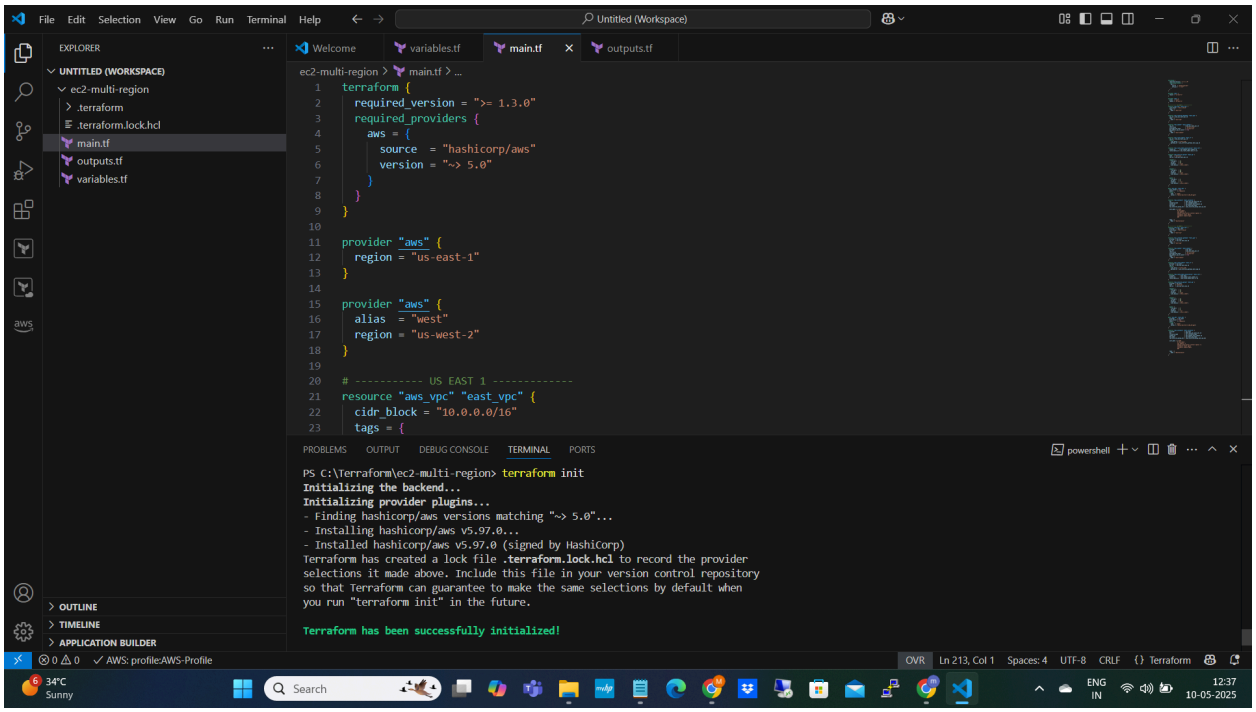
$ aws configure :



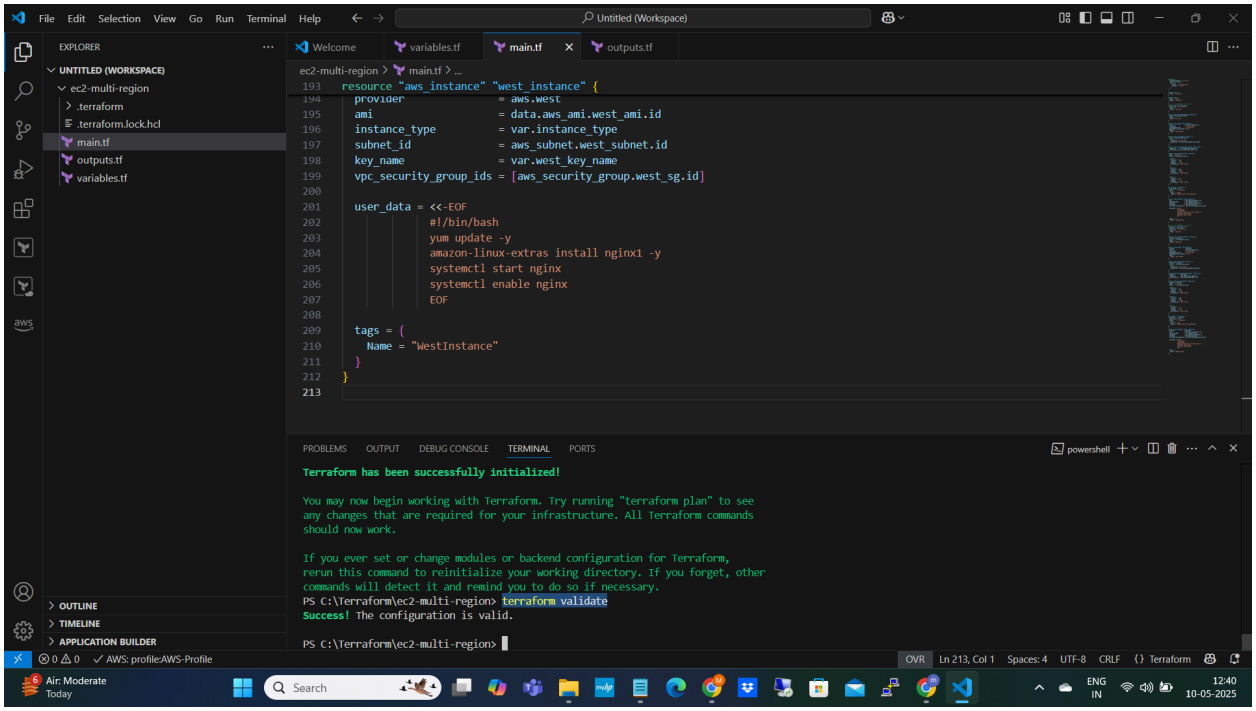
Access the directory : **$ cd ec2-multi-region**



Now Initialize the Terraform: **$ terraform init**

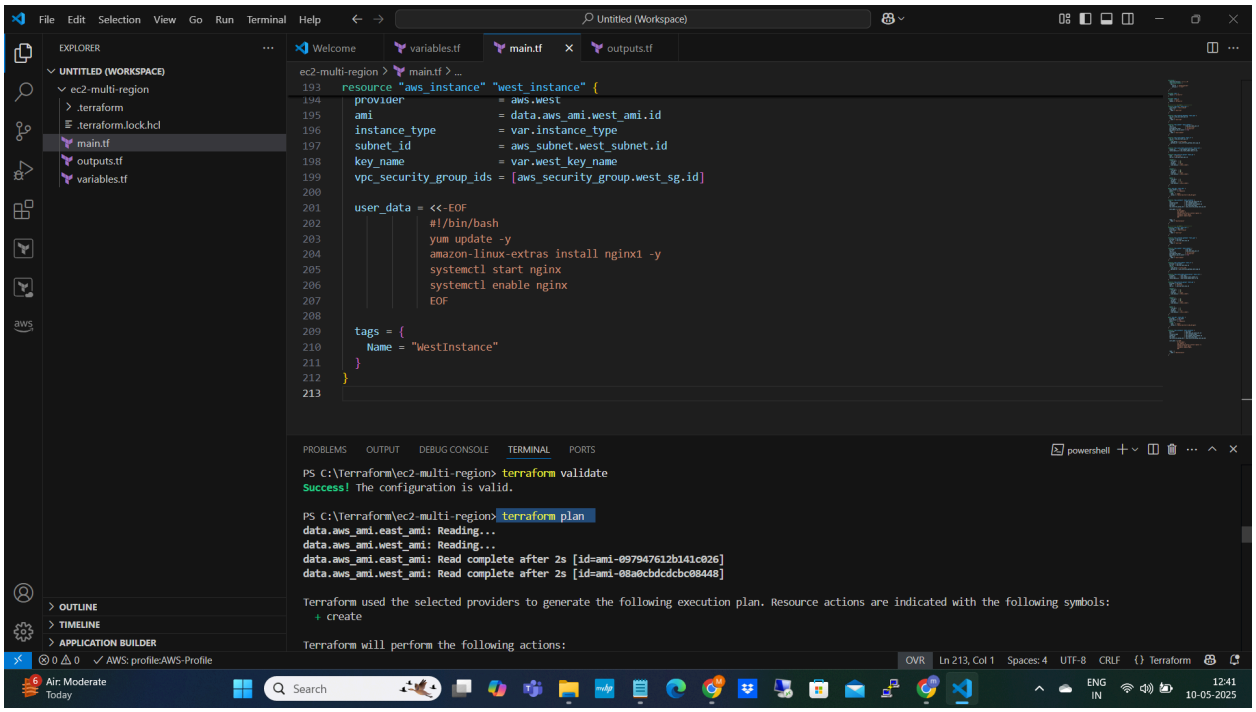


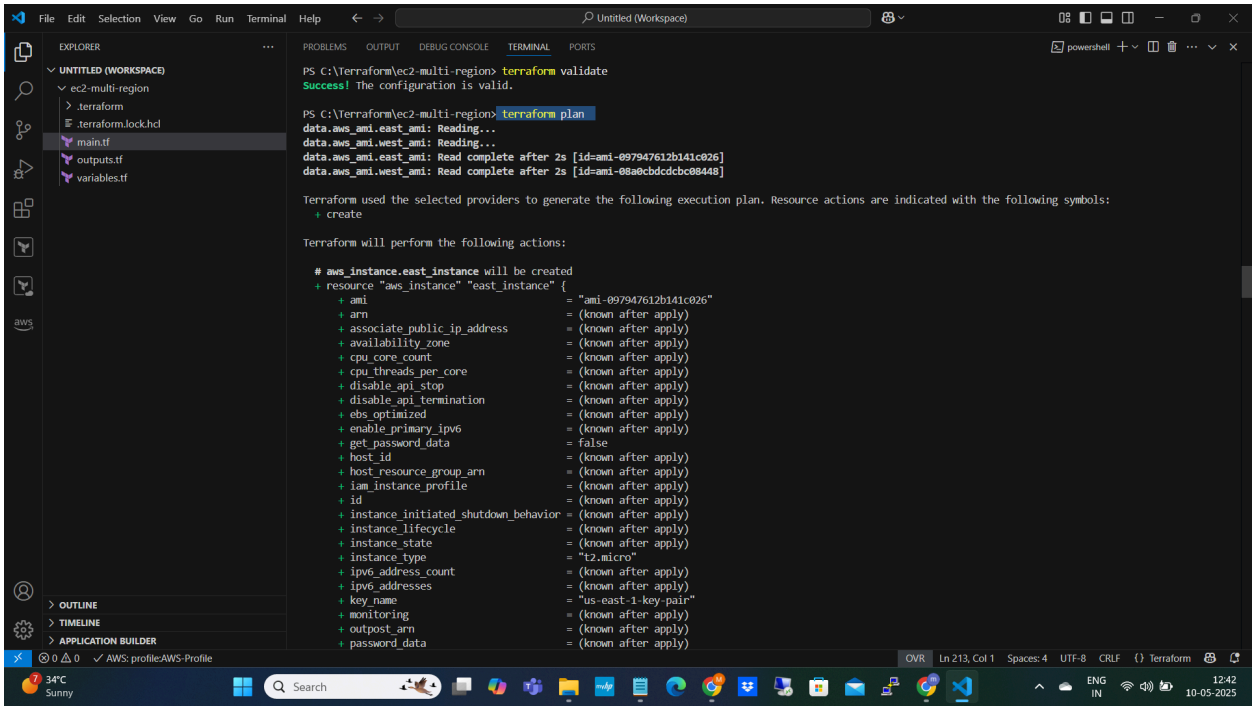
$ terraform validate

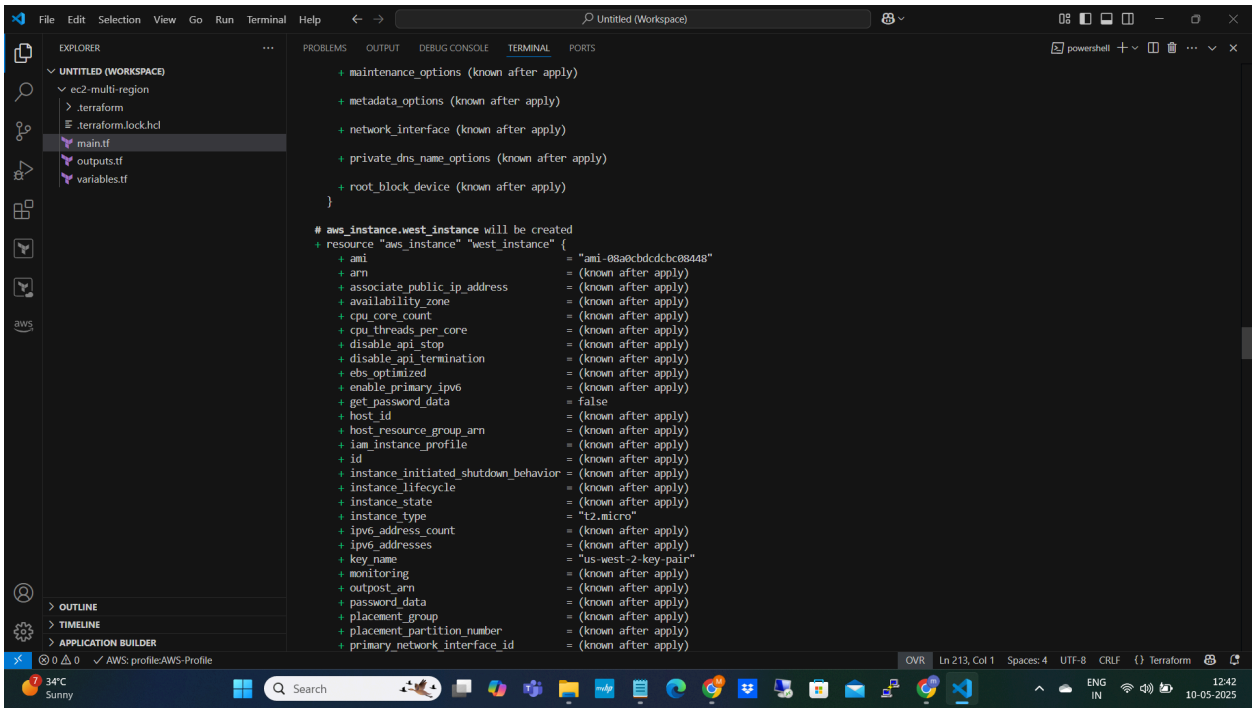


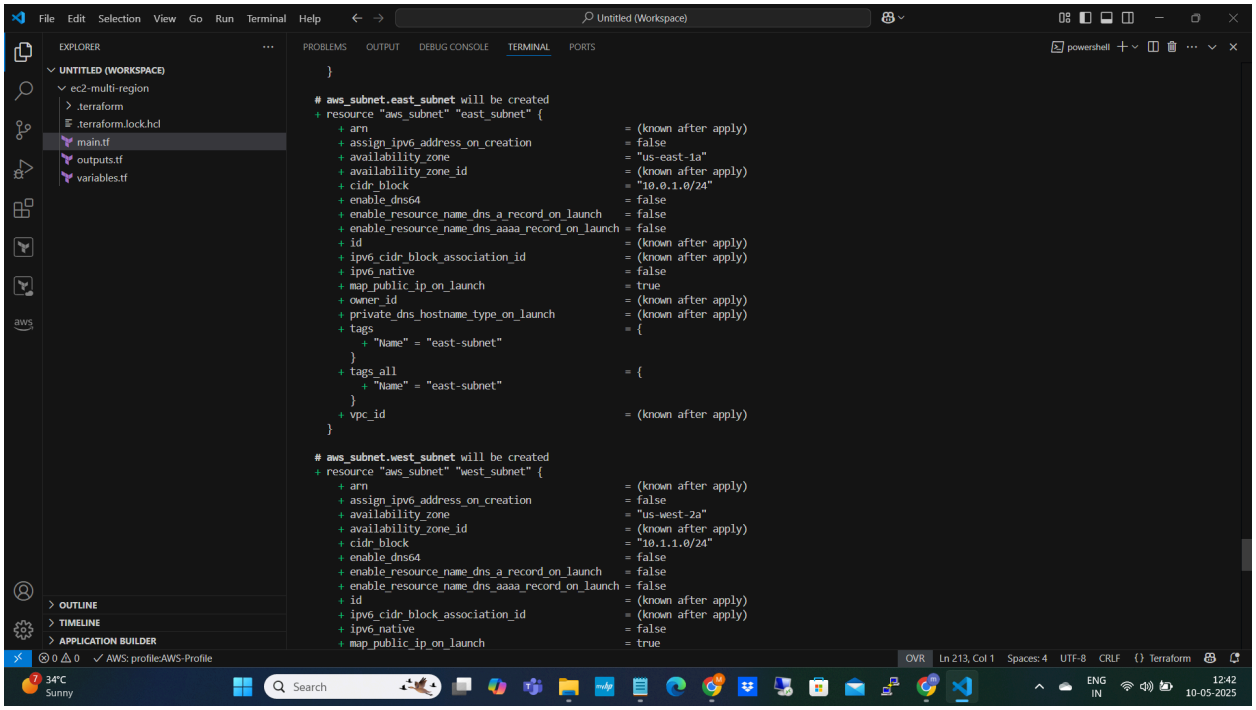
Lets display the plan before execution:

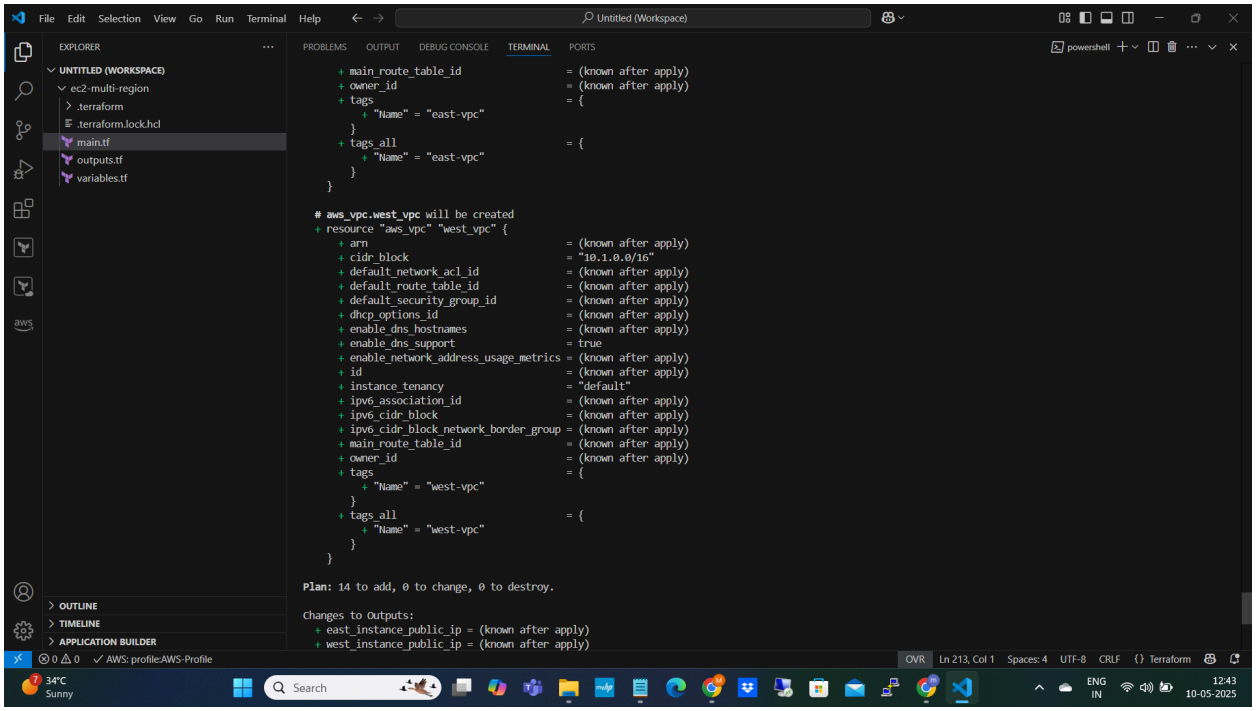
$ terraform plan




File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
ec2-multi-region
.terraform
.terraform.lock.hcl
main.tf
outputs.tf
variables.tf
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Terraform\ec2-multi-region> terraform validate
Success! The configuration is valid.
PS C:\Terraform\ec2-multi-region> terraform plan
data.aws_ami.east_ami: Reading...
data.aws_ami.west_ami: Reading...
data.aws_ami.east_ami: Read complete after 2s [id=ami-097947612b141c026]
data.aws_ami.west_ami: Read complete after 2s [id=ami-08a0cbdcdcbc08448]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.east_instance will be created
+ resource "aws_instance" "east_instance" {
+ ami = "ami-097947612b141c026"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "us-east-1-key-pair"
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
OUTLINE
TIMELINE
APPLICATION BUILDER
AWS: profile:AWS-Profile
OVR Ln 213, Col 1 Spaces: 4 UTF-8 CRLF {} Terraform
34°C Sunny
Search
ENG IN
12:42 10-05-2025

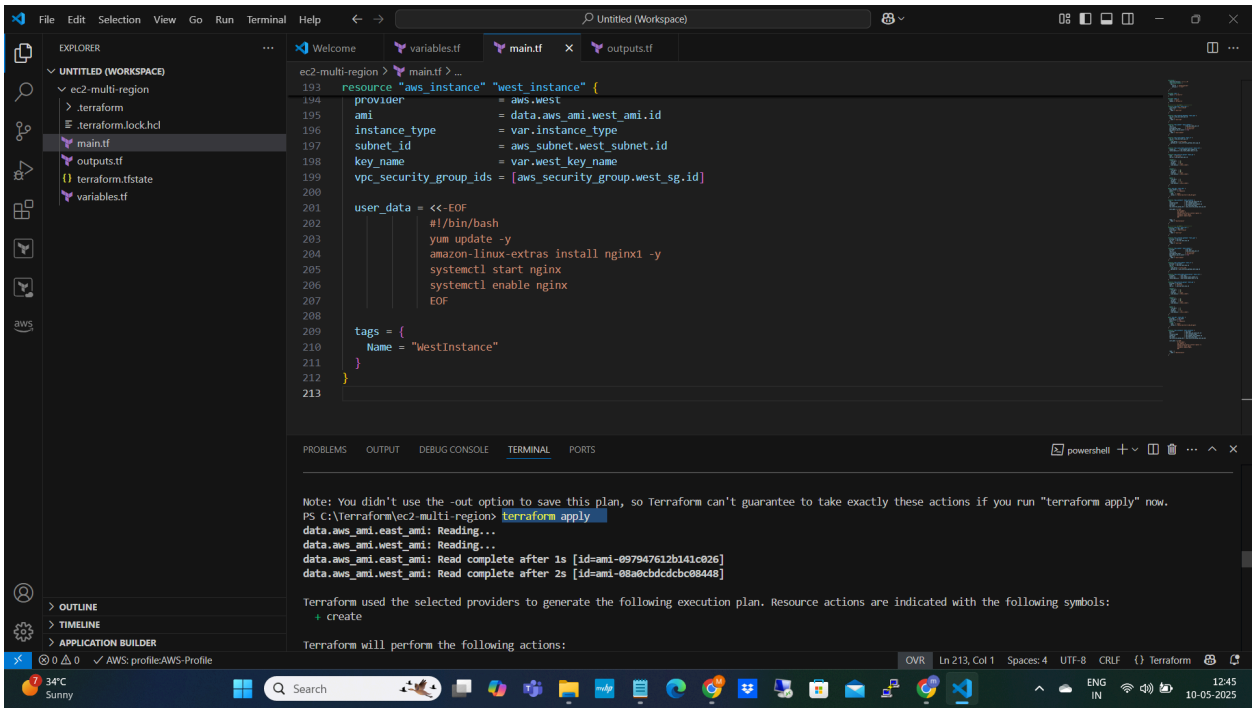

File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
ec2-multi-region
.terraform
.terraform.lock.hcl
main.tf
outputs.tf
variables.tf
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
# aws_instance.west_instance will be created
+ resource "aws_instance" "west_instance" {
+ ami = "ami-08a0cbdcdcbc08448"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ enable_primary_ipv6 = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "us-west-2-key-pair"
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
OUTLINE
TIMELINE
APPLICATION BUILDER
AWS: profile:AWS-Profile
OVR Ln 213, Col 1 Spaces: 4 UTF-8 CRLF {} Terraform
34°C Sunny
Search
ENG IN
12:42 10-05-2025

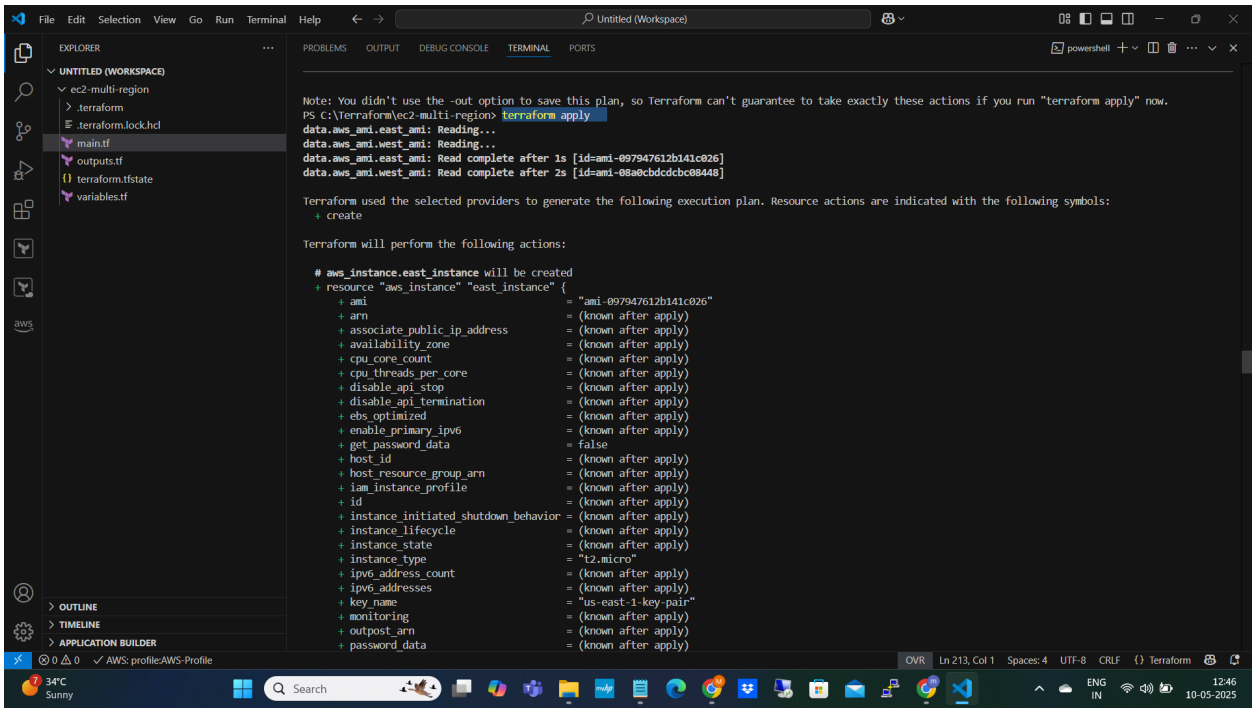

File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
ec2-multi-region
.terraform
.terraform.lock.hcl
main.tf
outputs.tf
variables.tf
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
}
# aws_subnet.east_subnet will be created
+ resource "aws_subnet" "east_subnet" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-east-1a"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.0.1.0/24"
+ enable_dns64 = false
+ enable_resource_name_dns_a_record_on_launch = false
+ enable_resource_name_dns_aaaa_record_on_launch = false
+ id = (known after apply)
+ ipv6_cidr_block_association_id = (known after apply)
+ ipv6_native = false
+ map_public_ip_on_launch = true
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags = {
+ "Name" = "east-subnet"
}
+ tags_all = {
+ "Name" = "east-subnet"
}
+ vpc_id = (known after apply)
}
# aws_subnet.west_subnet will be created
+ resource "aws_subnet" "west_subnet" {
+ arn = (known after apply)
+ assign_ipv6_address_on_creation = false
+ availability_zone = "us-west-2a"
+ availability_zone_id = (known after apply)
+ cidr_block = "10.1.1.0/24"
+ enable_dns64 = false
+ enable_resource_name_dns_a_record_on_launch = false
+ enable_resource_name_dns_aaaa_record_on_launch = false
+ id = (known after apply)
+ ipv6_cidr_block_association_id = (known after apply)
+ ipv6_native = false
+ map_public_ip_on_launch = true
OUTLINE
TIMELINE
APPLICATION BUILDER
AWS: profile:AWS-Profile
OVR Ln 213, Col 1 Spaces: 4 UTF-8 CRLF {} Terraform
34°C Sunny
Search
ENG IN
12:42 10-05-2025


File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
ec2-multi-region
.terraform
.terraform.lock.hcl
main.tf
outputs.tf
variables.tf
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
+ tags = {
+ "Name" = "east-vpc"
}
+ tags_all = {
+ "Name" = "east-vpc"
}
}
# aws_vpc.west_vpc will be created
+ resource "aws_vpc" "west_vpc" {
+ arn = (known after apply)
+ cidr_block = "10.1.0.0/16"
+ default_network_acl_id = (known after apply)
+ default_route_table_id = (known after apply)
+ default_security_group_id = (known after apply)
+ dhcp_options_id = (known after apply)
+ enable_dns_hostnames = (known after apply)
+ enable_dns_support = true
+ enable_network_address_usage_metrics = (known after apply)
+ id = (known after apply)
+ instance_tenancy = "default"
+ ipv6_association_id = (known after apply)
+ ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
+ tags = {
+ "Name" = "west-vpc"
}
+ tags_all = {
+ "Name" = "west-vpc"
}
}
Plan: 14 to add, 0 to change, 0 to destroy.
Changes to Outputs:
+ east_instance_public_ip = (known after apply)
+ west_instance_public_ip = (known after apply)
OUTLINE
TIMELINE
APPLICATION BUILDER
AWS: profile:AWS-Profile
OVR Ln 213, Col 1 Spaces: 4 UTF-8 CRLF {} Terraform
34°C Sunny
Search
ENG IN
12:43 10-05-2025

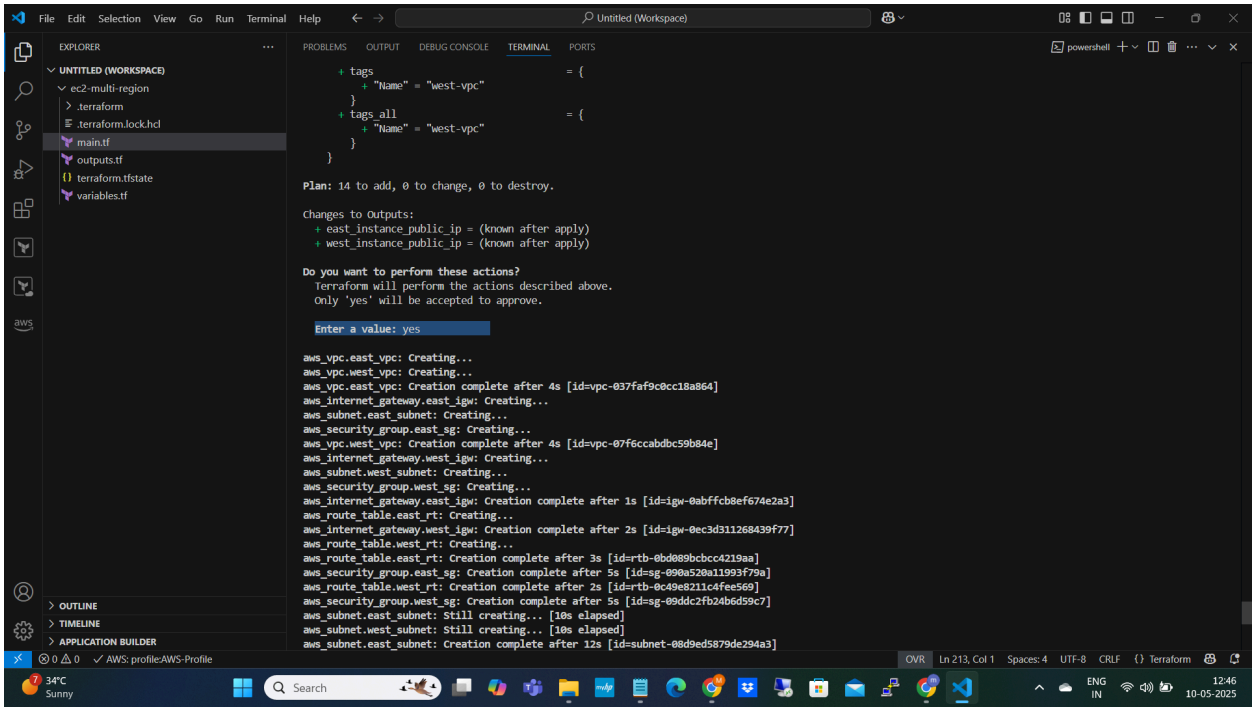
Now Execute the plan by executing apply command

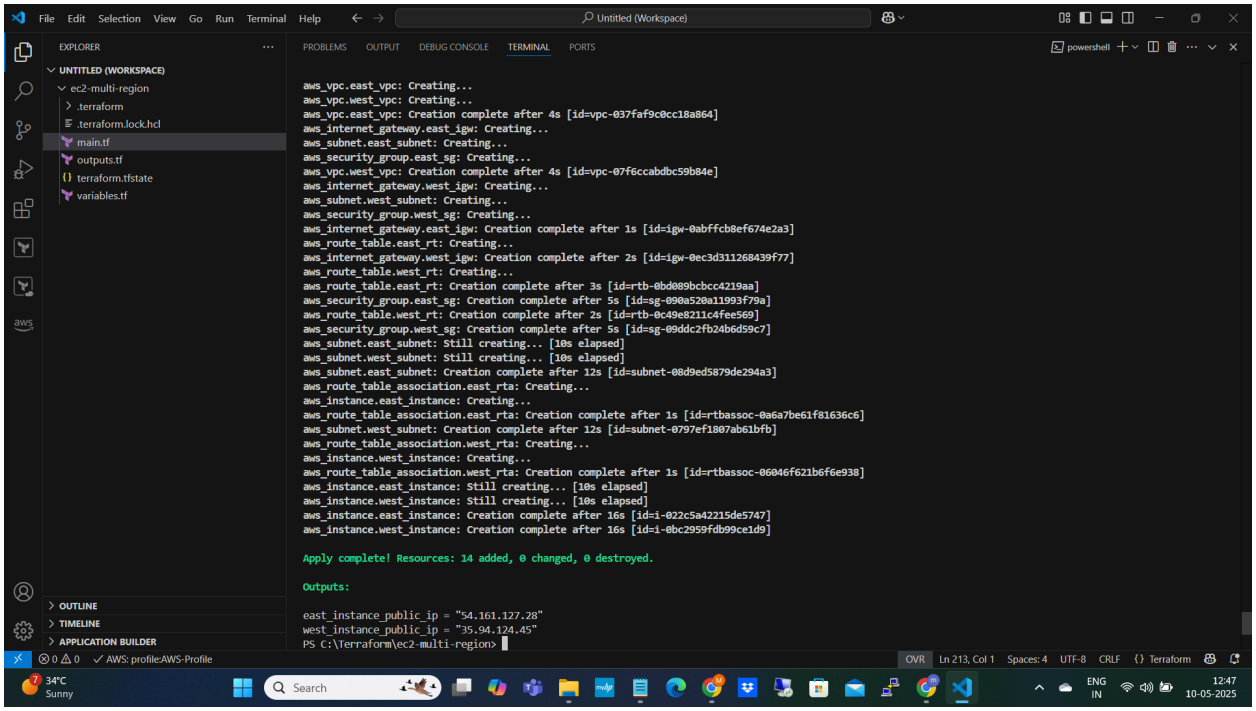
$ terraform apply





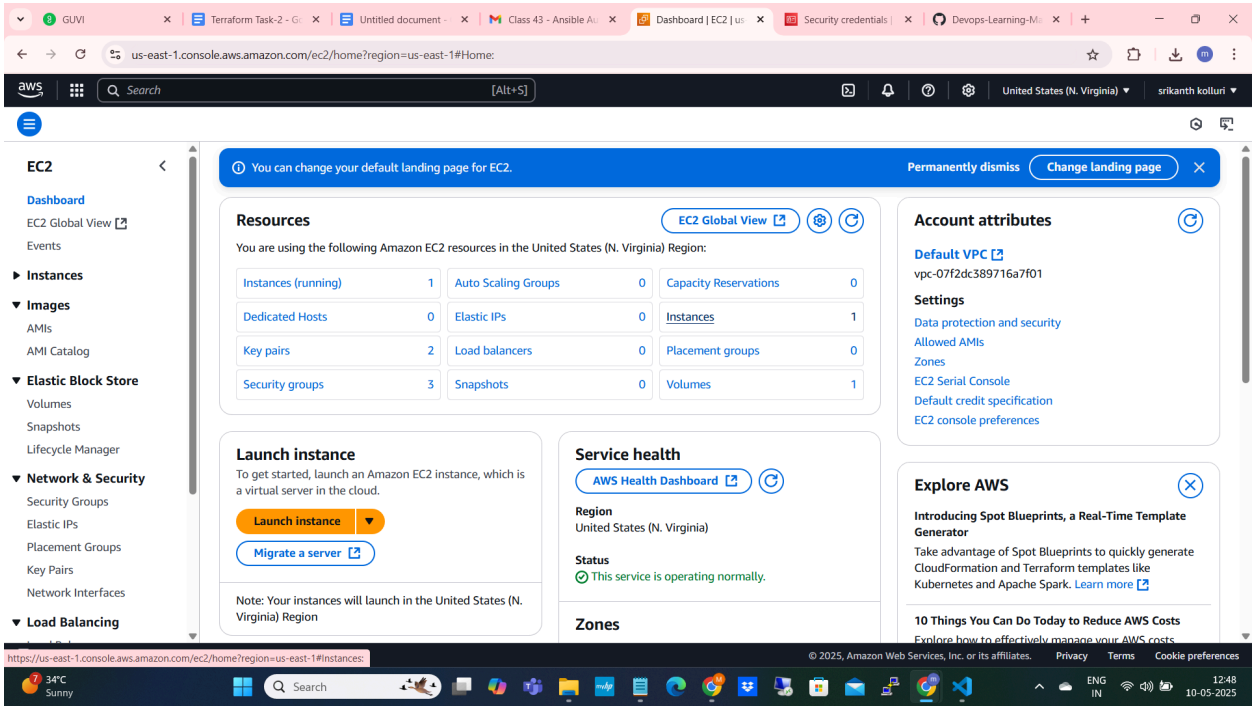
Entered Value as : yes to execute apply



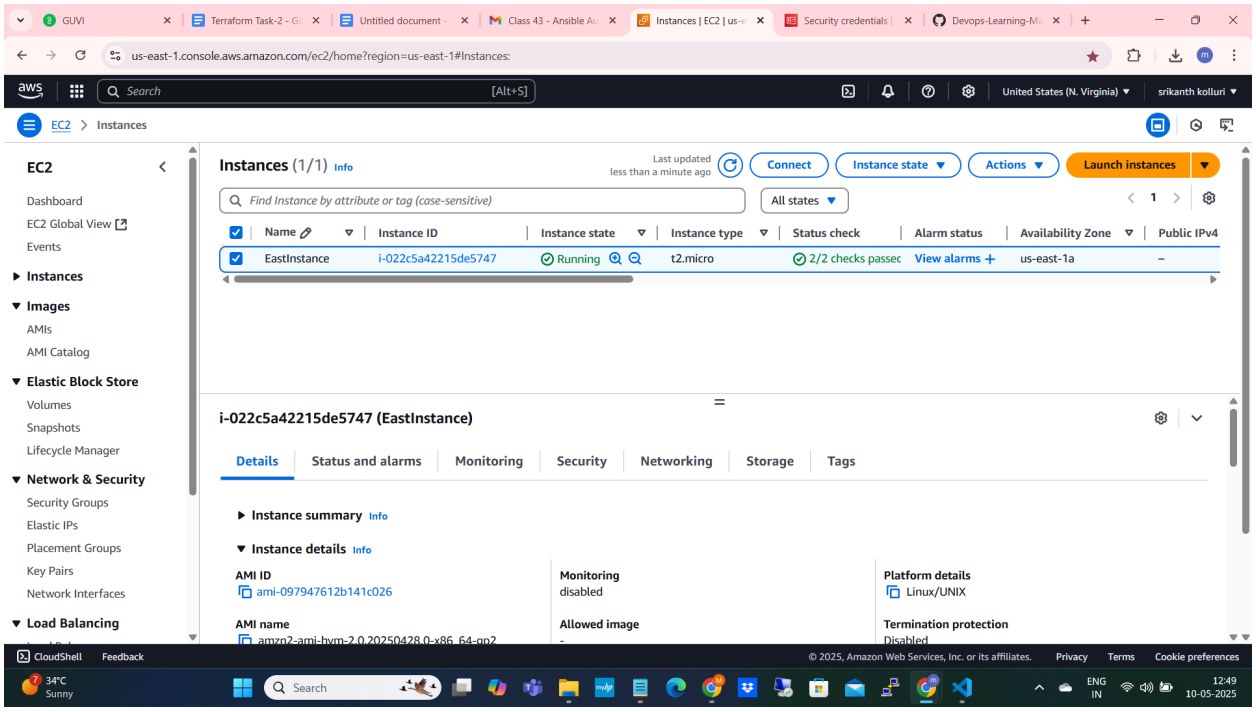


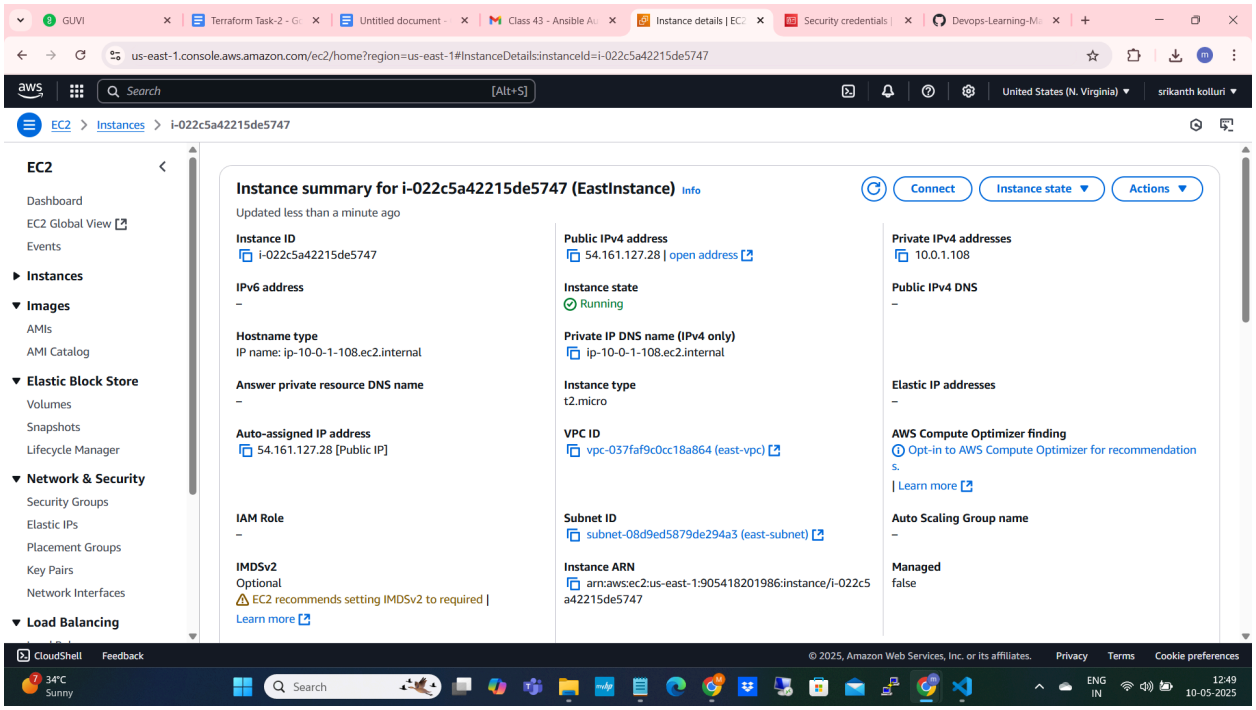
Apply execution done successfully

Now let's check the AWS console whether the 2 Ec2 instances created in two regions:



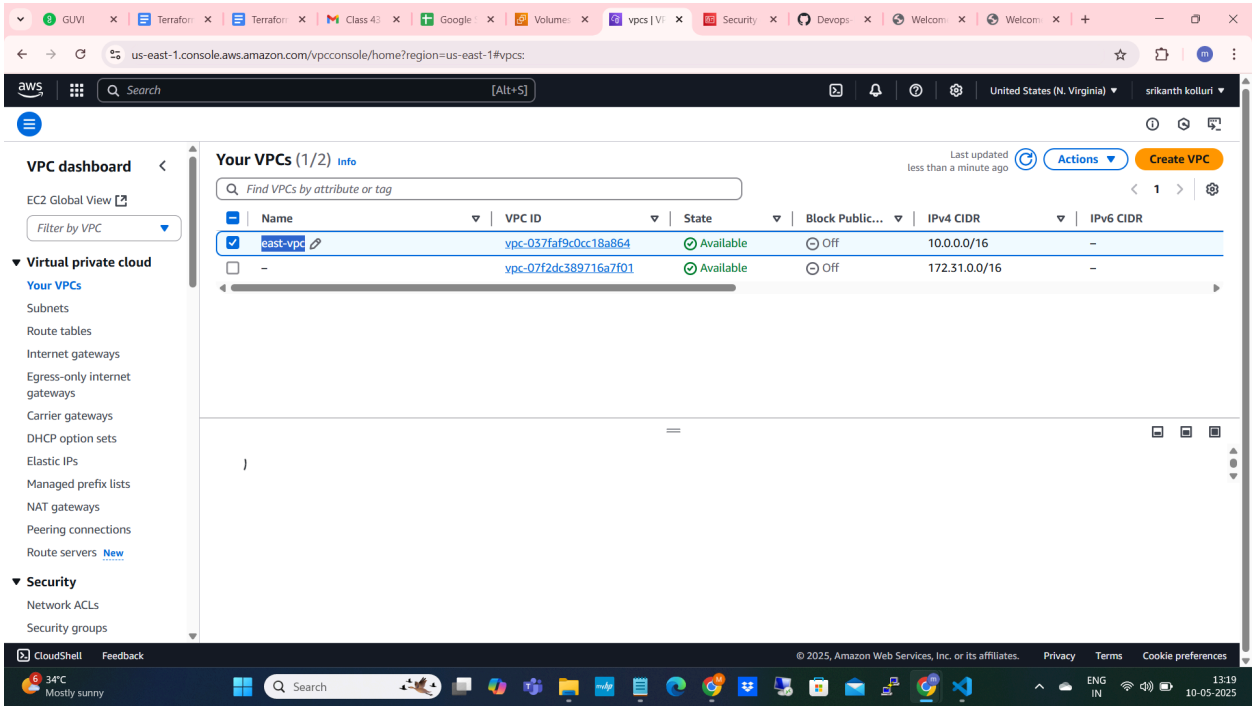
EastInstance created in us-east-1 region:

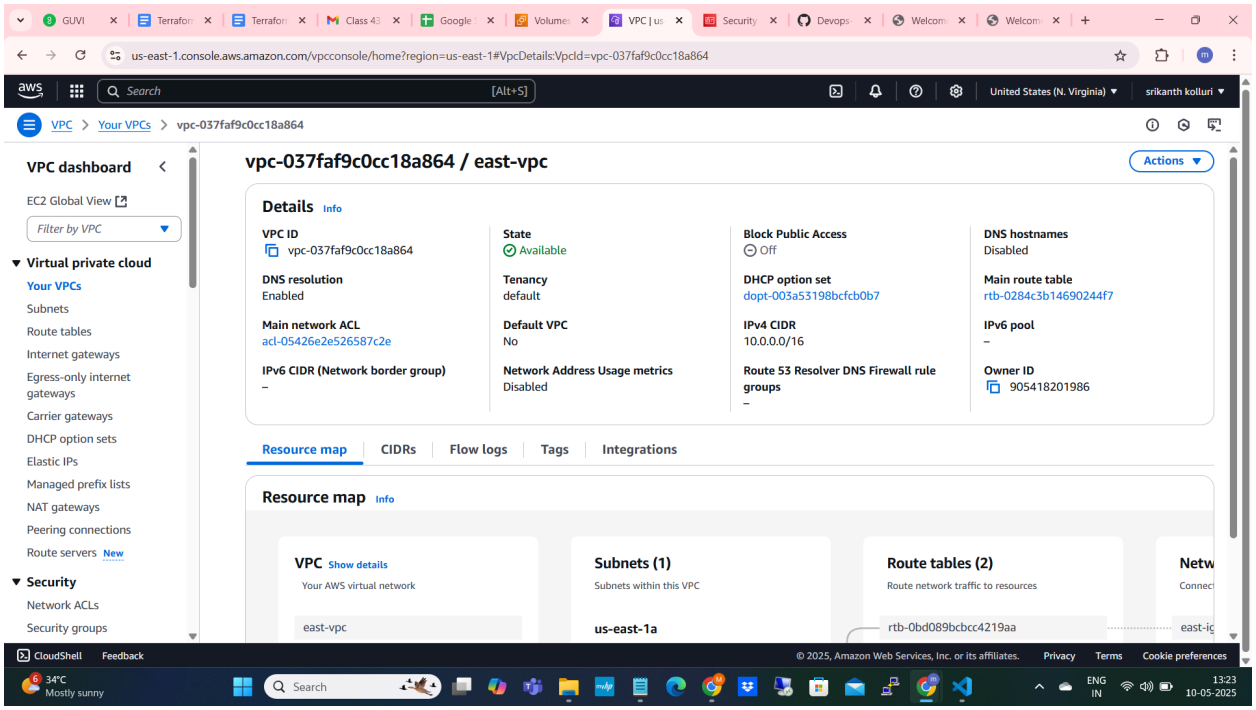




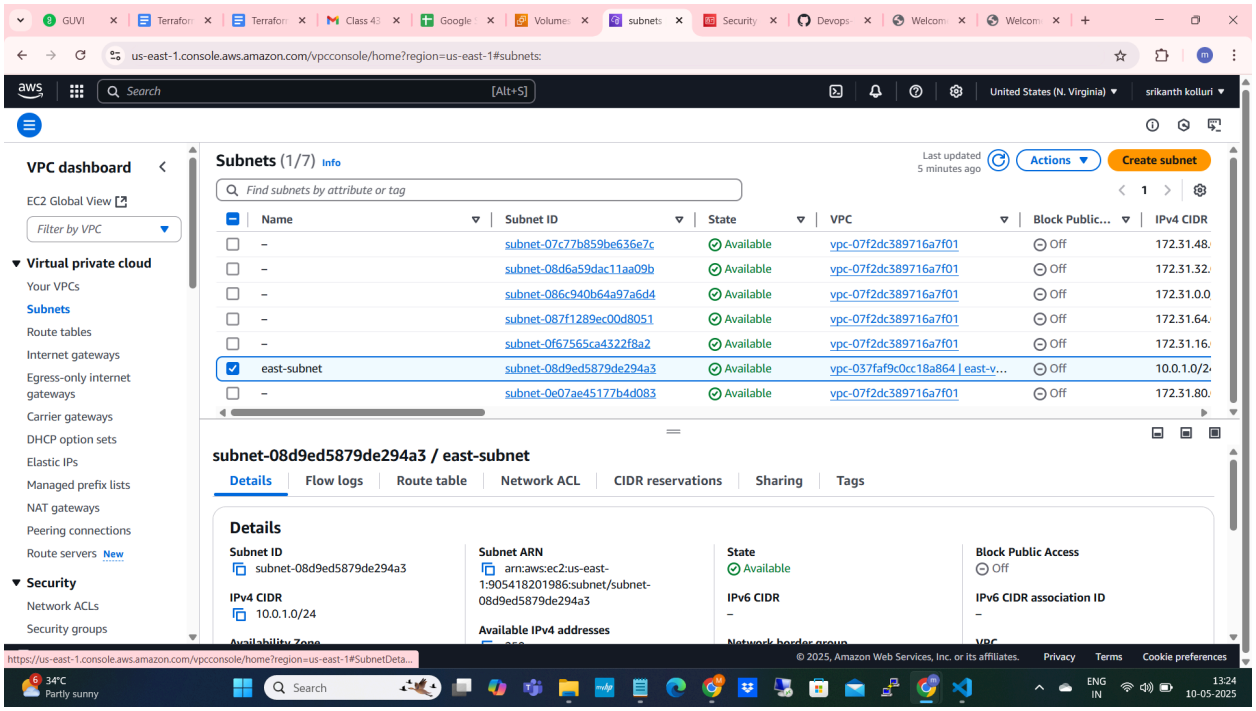
us-east-1: Vpc,subnets and Internet gateway:

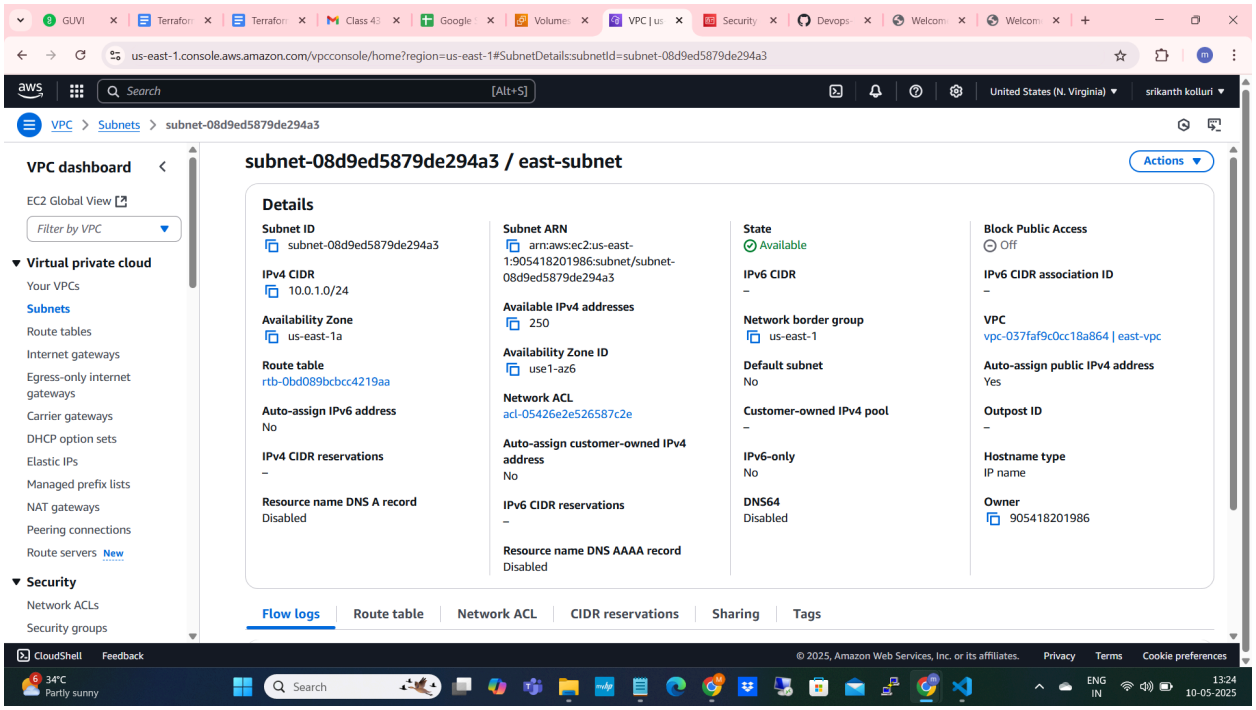
vpc:



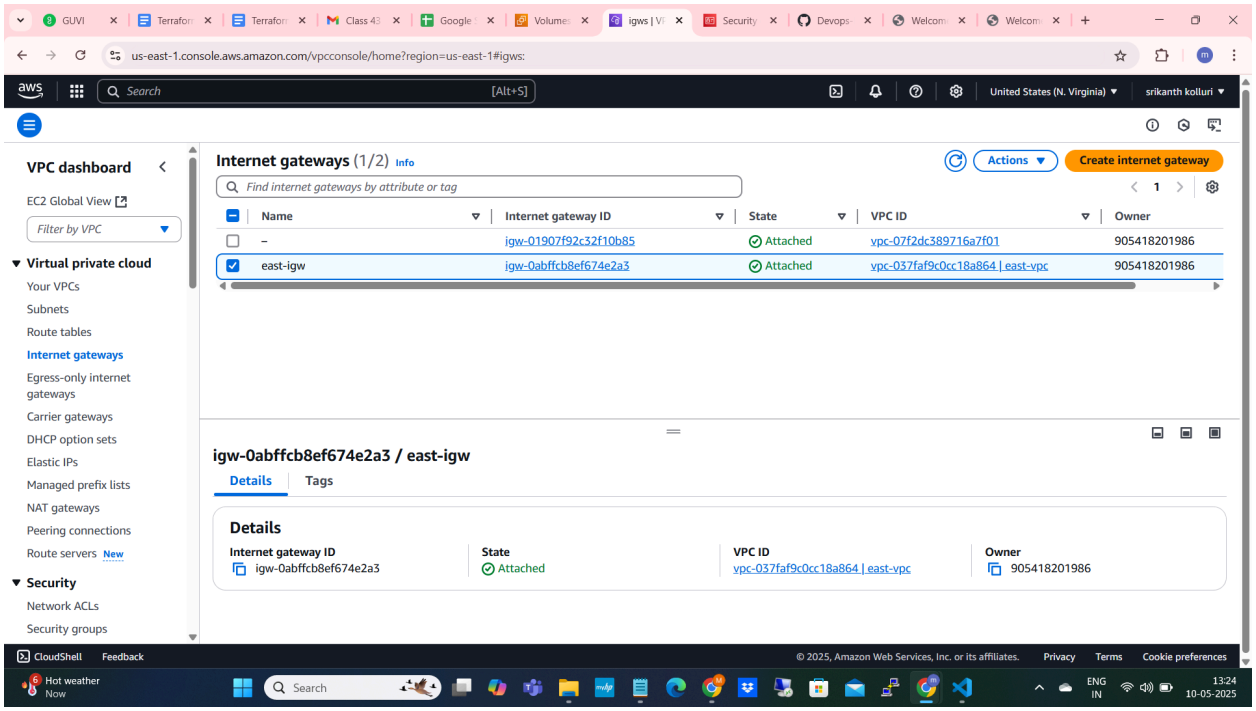

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails:VpcId=vpc-037faf9c0cc18a864
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
vpc-037faf9c0cc18a864 / east-vpc
Actions
Details Info
VPC ID
vpc-037faf9c0cc18a864
State
Available
Block Public Access
Off
DNS hostnames
Disabled
DNS resolution
Enabled
Tenancy
default
DHCP option set
dopt-003a53198bcfcb0b7
Main route table
rtb-0284c3b14690244f7
Main network ACL
acl-05426e2e526587c2e
Default VPC
No
IPv4 CIDR
10.0.0.0/16
IPv6 pool
–
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
905418201986
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
east-vpc
Subnets (1)
Subnets within this VPC
us-east-1a
Route tables (2)
Route network traffic to resources
rtb-0bd089bcbcc4219aa
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

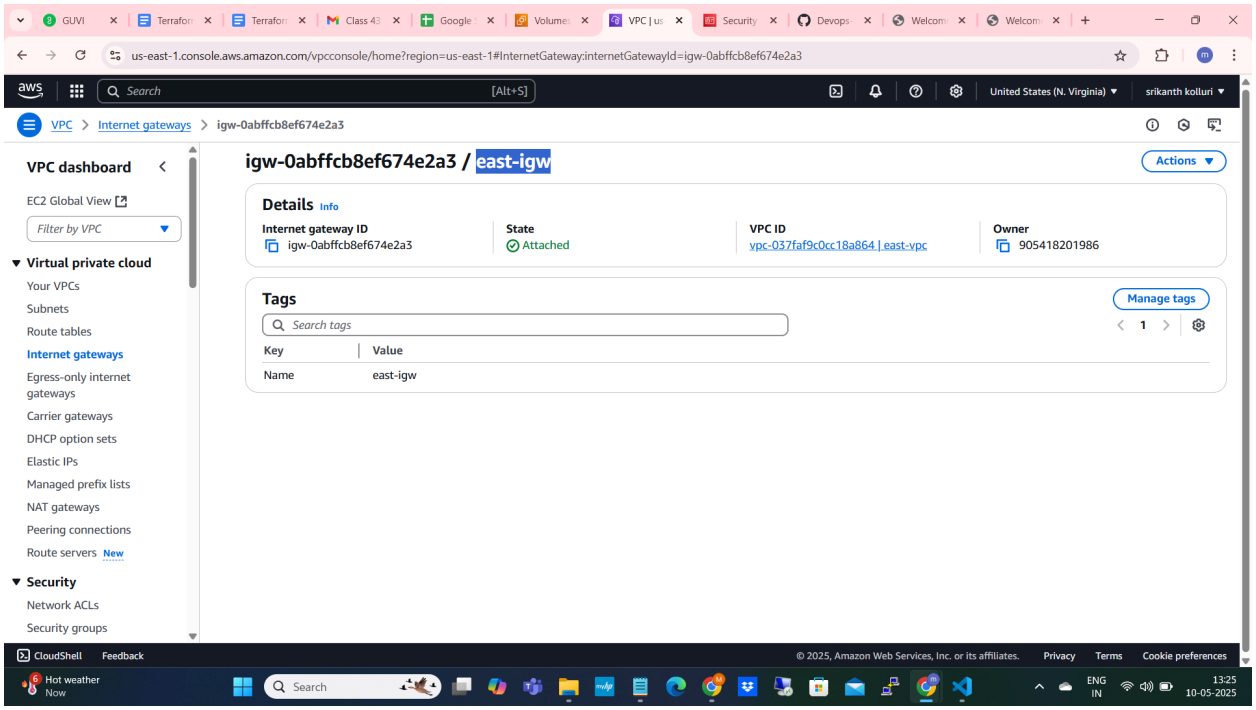
Subnet:


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets:
VPC dashboard
Subnets (1/7) Info
Last updated 5 minutes ago
Actions
Create subnet
Find subnets by attribute or tag
Name
Subnet ID
State
VPC
Block Public...
IPv4 CIDR
subnet-07c77b859be636e7c
subnet-08d6a59dac11aa09b
subnet-086c940b64a97a6d4
subnet-087f1289ec00d8051
subnet-0f67565ca4322f8a2
east-subnet
subnet-08d9ed5879de294a3
subnet-0e07ae45177b4d083
vpc-07f2dc389716a7f01
vpc-037faf9c0cc18a864 | east-v...
Available
Off
subnet-08d9ed5879de294a3 / east-subnet
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Subnet ID
subnet-08d9ed5879de294a3
Subnet ARN
arn:aws:ec2:us-east-1:905418201986:subnet/subnet-08d9ed5879de294a3
State
Available
Block Public Access
Off
IPv4 CIDR
10.0.1.0/24
IPv6 CIDR
–
IPv6 CIDR association ID
–
Available IPv4 addresses
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

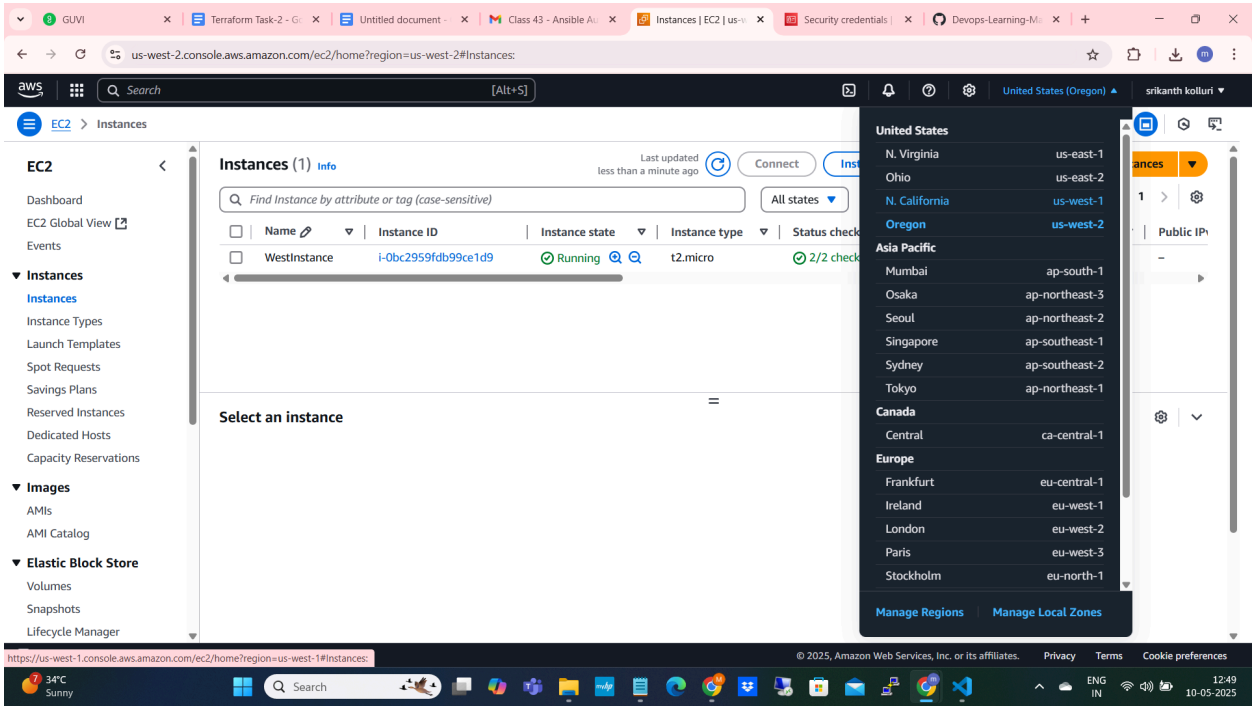


internet gateway:

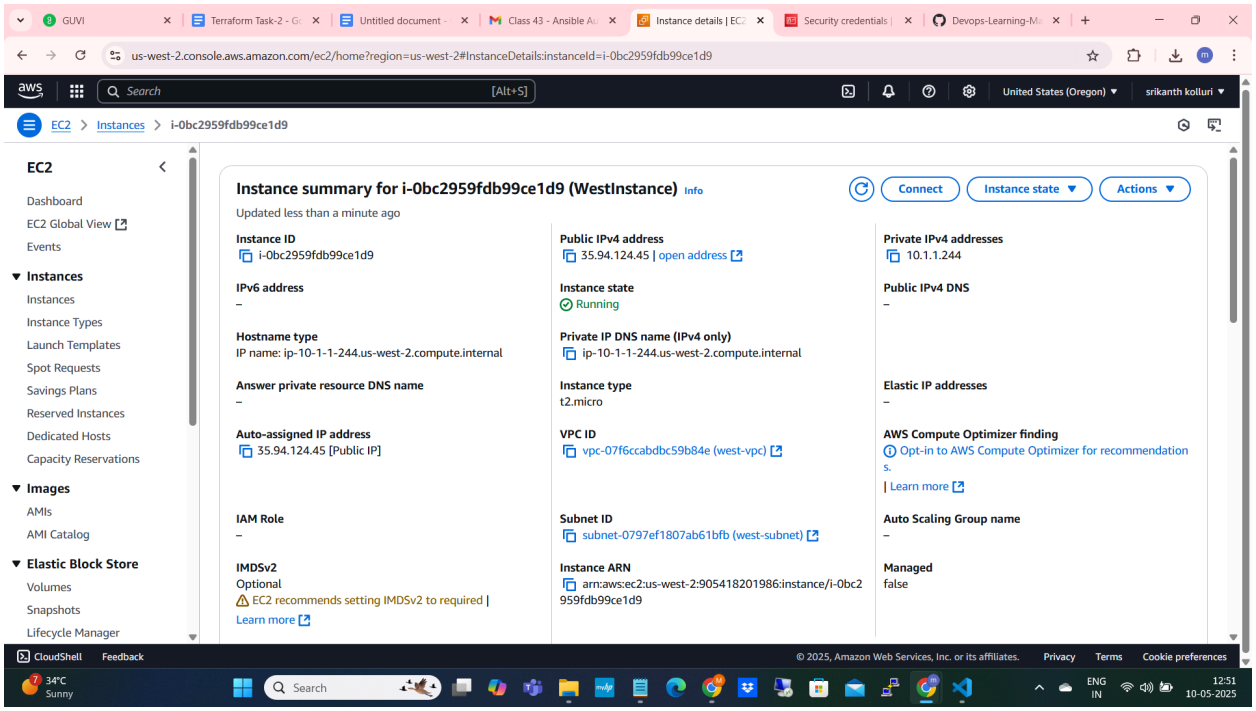




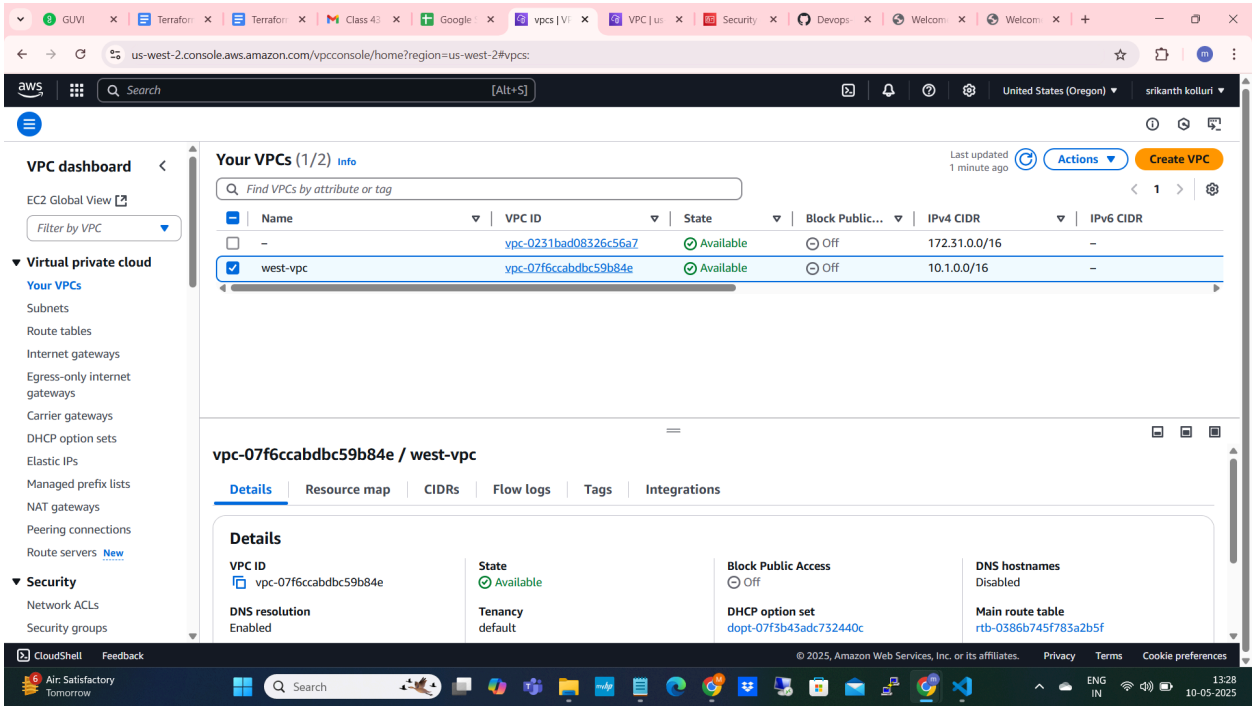
Changed the Region to us-west-2:

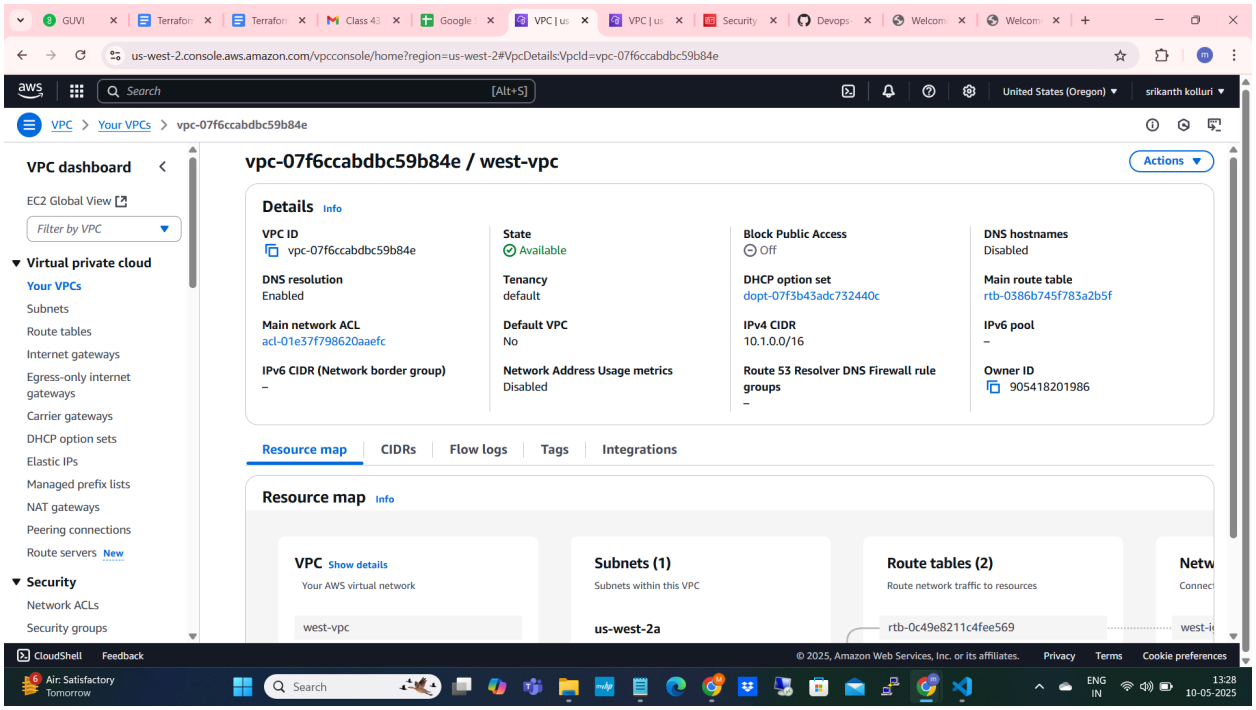


WestInstance created in us-west-2 region:

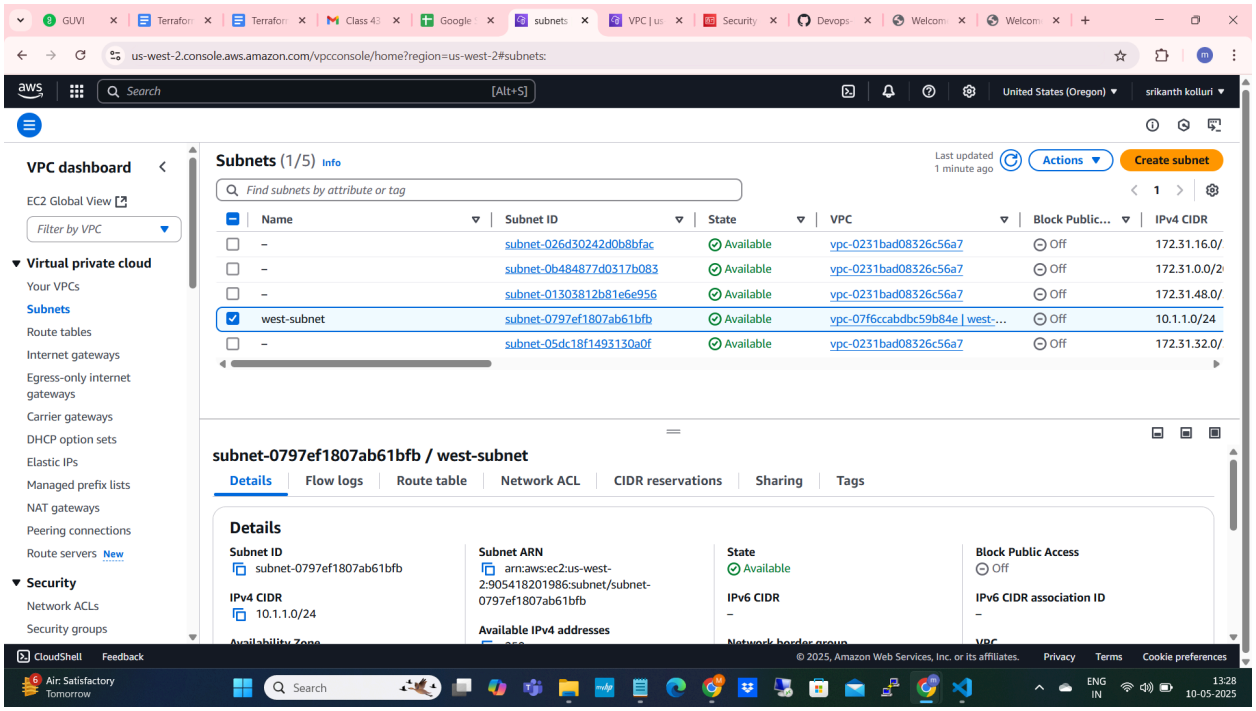


us-west-2: Vpc,subnets and Internet gateway:

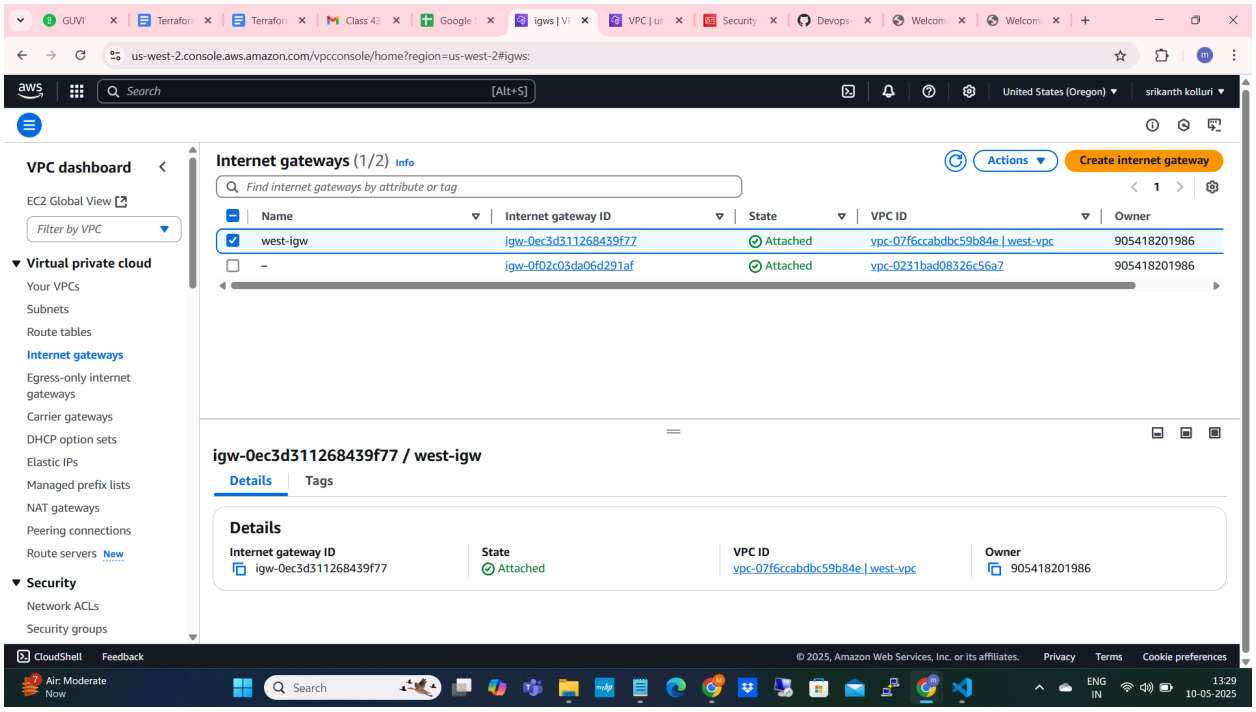


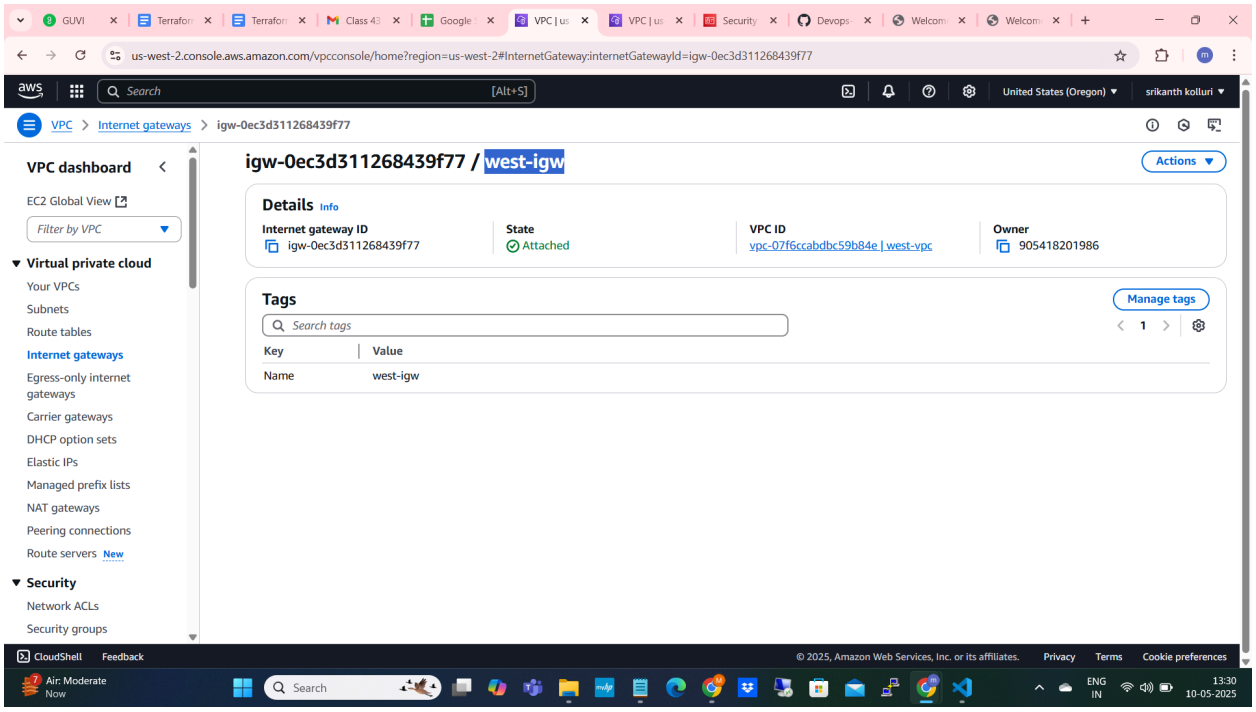

us-west-2.console.aws.amazon.com/vpcconsole/home?region=us-west-2#VpcDetails:VpcId=vpc-07f6ccabdbc59b84e
VPC > Your VPCs > vpc-07f6ccabdbc59b84e
VPC dashboard
vpc-07f6ccabdbc59b84e / west-vpc
Actions
Details Info
VPC ID
vpc-07f6ccabdbc59b84e
State
Available
Block Public Access
Off
DNS hostnames
Disabled
DNS resolution
Enabled
Tenancy
default
DHCP option set
dopt-07f3b43adc732440c
Main route table
rtb-0386b745f783a2b5f
Main network ACL
acl-01e37f798620aaefc
Default VPC
No
IPv4 CIDR
10.1.0.0/16
IPv6 pool
–
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
905418201986
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
west-vpc
Subnets (1)
Subnets within this VPC
us-west-2a
Route tables (2)
Route network traffic to resources
rtb-0c49e8211c4fee569

Subnet:


us-west-2.console.aws.amazon.com/vpcconsole/home?region=us-west-2#subnets:
VPC dashboard
Subnets (1/5) Info
Last updated 1 minute ago
Actions
Create subnet
Find subnets by attribute or tag
Name
Subnet ID
State
VPC
Block Public...
IPv4 CIDR
subnet-026d30242d0b8bfac
subnet-0b484877d0317b083
subnet-01303812b81e6e956
west-subnet
subnet-0797ef1807ab61bfb
subnet-05dc18f1493130a0f
subnet-0797ef1807ab61bfb / west-subnet
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Details
Subnet ID
subnet-0797ef1807ab61bfb
Subnet ARN
arn:aws:ec2:us-west-2:905418201986:subnet/subnet-0797ef1807ab61bfb
State
Available
Block Public Access
Off
IPv4 CIDR
10.1.1.0/24
IPv6 CIDR
–
IPv6 CIDR association ID
–
Available IPv4 addresses

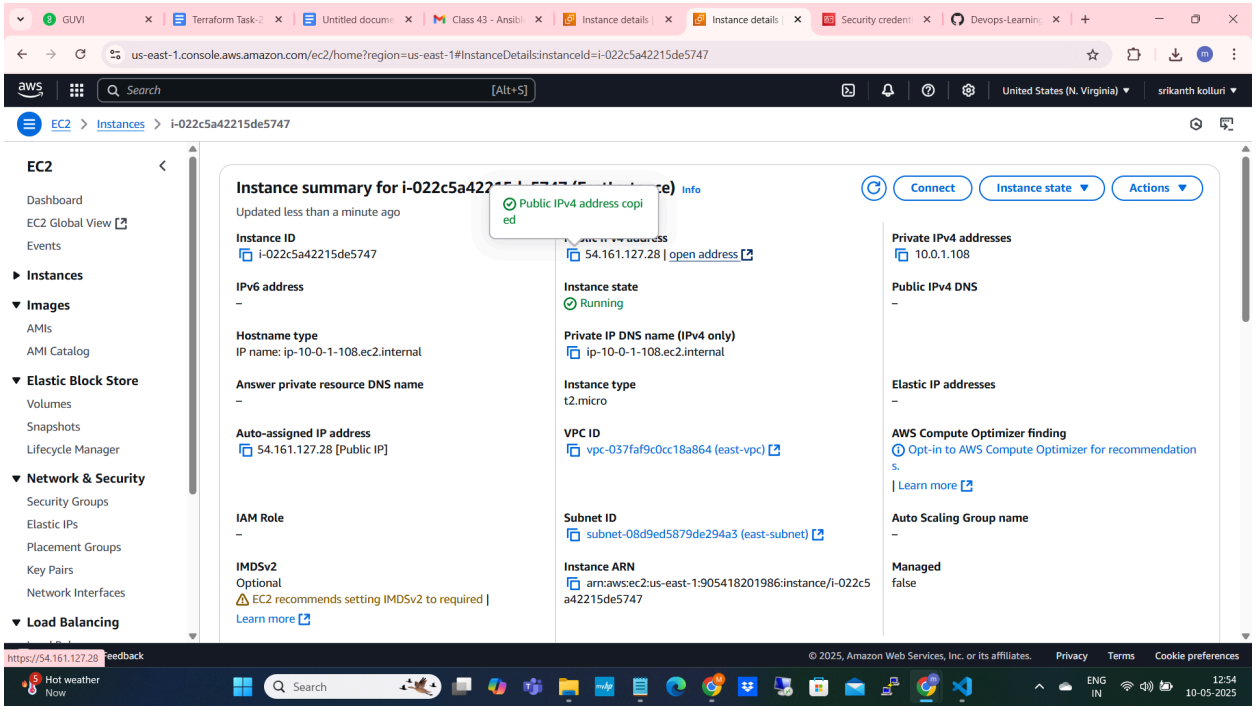
Internet Gateway:



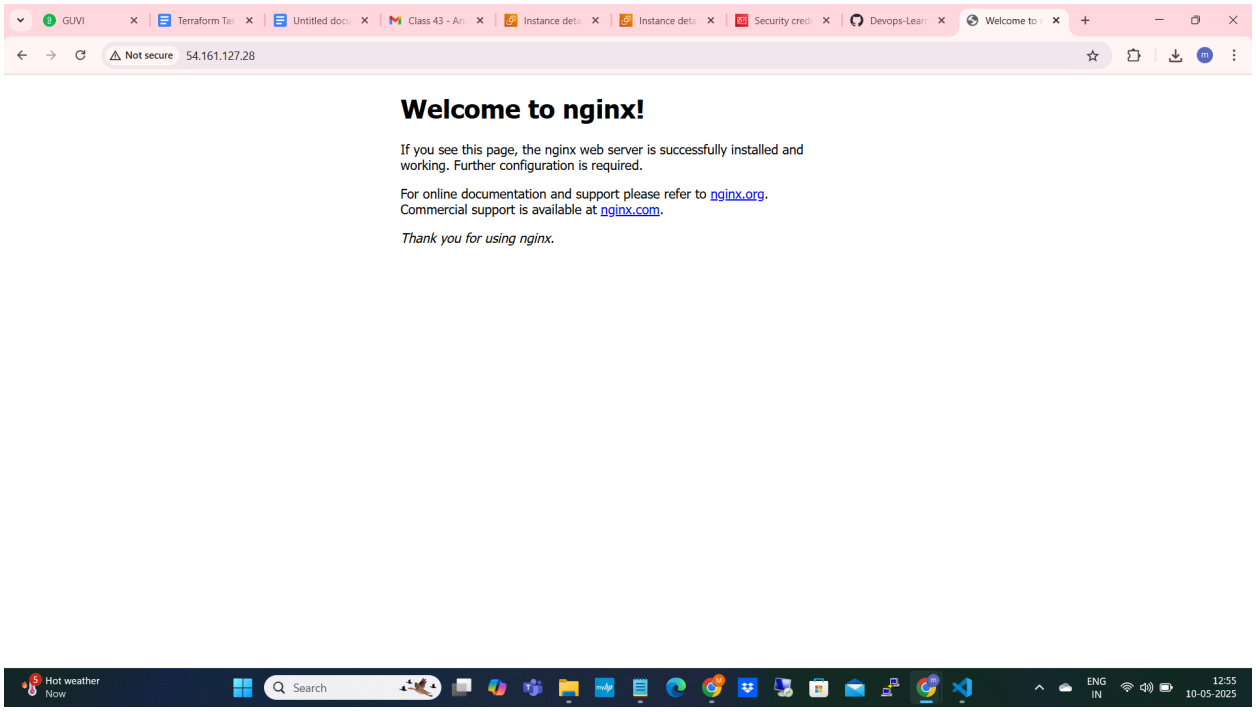


Now lets check whether the Nginx installed in both instances by accessing them:

Us-east-1 : ip : 54.161.127.28

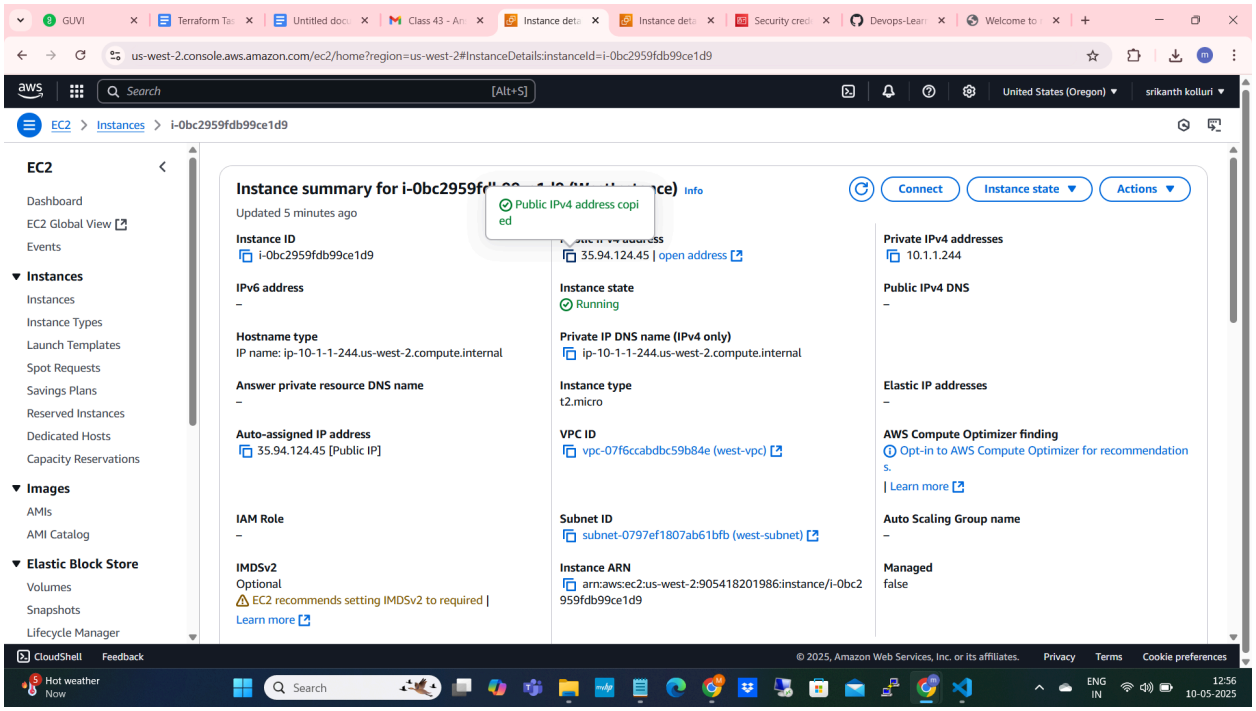


http://54.161.127.28/

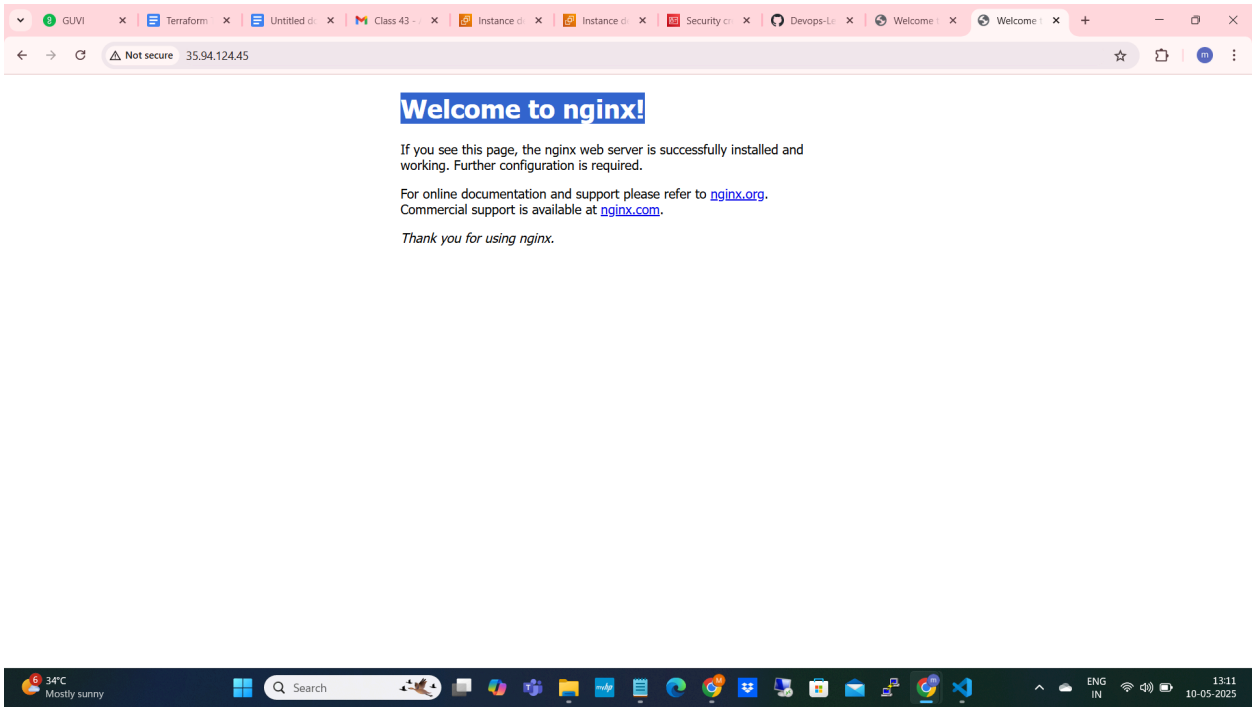


Nginx Installed in us-east-1 region instance

Us-west-2 : ip : 35.94.124.45



http://35.94.124.45/



Nginx Installed in us-west-2 region instance